



Bruno Miguel Guerra Duarte Galveia

Licenciado em Ciências de Engenharia Eletrotécnica e de Computadores

Extensão do SDK do Kinect: Criação de uma Biblioteca de Gestos

Dissertação para obtenção do Grau de Mestre em Engenharia Eletrotécnica e de Computadores

Orientador: Tiago Cardoso, Professor Auxiliar, FCT-UNL

Co-orientador: Yves Rybarczyk, Professor Auxiliar, FCT-UNL

Júri:

Presidente: Prof. Doutor João Miguel Murta Pina

Arguente: Prof. Doutor Vítor Manuel Pereira Duarte dos Santos

Vogal: Prof. Doutor José António Barata de Oliveira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2014

Copyright

Extensão do SDK do Kinect: Criação de uma Biblioteca de Gestos

**Bruno Miguel Guerra Duarte Galveia, FCT/UNL, UNL Copyright ©.
Todos os direitos reservados.**

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Agradecimentos

Gostaria de agradecer ao meu orientador Prof. Tiago Cardoso pela oportunidade, pelo apoio e principalmente pela motivação ao longo desta dissertação. Ao meu co-orientador Prof. Yves Rybarczyk.

À Universidade Nova de Lisboa e a Faculdade de Ciências de Tecnologias por me terem acolhido e disponibilizado as condições que permitiram a minha formação. Esta casa permitiu-me crescer a todos os níveis, deu a conhecer muitas pessoas e travar grandes amizades. Um obrigado também a todos os docentes que contribuíram para a minha formação. A todos os meus colegas de faculdade que me acompanharam, em especial ao eterno colega de casa João Silva, Bruno Vilhena, Nuno Vilhena, Marcos Carvalho, ao André Reduto pelo incentivo e companhia na realização desta dissertação e ao João Gameiro pelas dicas e colaboração.

Aos meus pais, Fernando e Joaquina, pelo apoio e pela possibilidade do percurso académico, agradeço também ao meu irmão, pelo apoio técnico e experiência na área, e a todos os outros familiares.

Por fim, um agradecimento aos meus amigos que me acompanham desde sempre pelos grandes momentos que passamos juntos e por estarem sempre lá.

Resumo

As interfaces naturais têm ganho atenção crescente por parte da comunidade científica. Esta interação, nomeadamente baseada em gestos e voz, tem sido progressivamente mais usada para estabelecer a comunicação entre o homem e a máquina de forma natural. Em paralelo, têm surgido no mercado alguns novos dispositivos com o objetivo de suportar estas formas naturais de interface, entre eles, sensores de movimento.

Esta dissertação tem como objetivo a proposta de algoritmos projetados para auxiliar o programador no reconhecimento de gestos, especificamente em dispositivos conhecidos como sensores de movimentos. Atualmente o programador tem o trabalho de desenvolver todo o código para adicionar reconhecimento de gestos à sua aplicação. A solução apresentada neste documento propõe um novo conjunto de métodos que visam a criação de uma biblioteca de gestos e a consequente capacidade de reconhecimento de gestos anteriormente armazenados.

A validação deste trabalho consistiu na extensão do *Software Development Kit* (SDK) do dispositivo Kinect, de forma a adicionar ferramentas complementares, que permitam simplificar a definição e reconhecimento de gestos. A primeira parte consiste na captação e armazenamento de gestos, através do mapeamento de pontos, modelando os gestos e armazenando tal modelo. Esta primeira fase permite no desenvolvimento de um sistema a criação de uma biblioteca de gestos pré-definidos, como por exemplo os gestos da Língua Gestual Portuguesa. Numa segunda fase, a extensão do SDK auxilia também as tarefas de reconhecimento de gestos em tempo real, isto é, deteção do gesto que o utilizador de uma aplicação esteja a fazer e procura desse gesto na biblioteca.

Palavras chave:

Extensão SDK

Biblioteca de gestos

Gravação de gestos

Reconhecimento de gestos em tempo real

Kinect

Abstract

Natural Interfaces has been steadily growing within the scientific community. Often making use of gestures and voice, it can be used to facilitate a natural interaction between man and machine. This potential motivates the development of novelty interaction devices, such as motion sensors, which can further support natural interaction.

This thesis work focuses on the development of algorithms designed to facilitate the recognition of human gestures through the use of motion sensors. Nowadays, it is common for a developer to generate all tools used for gesture recognition. The solution depicted in the work at hand proposes a new set of methods which aim at generating a library of gestures which can be used on diverse recognition applications.

Validation of the methods hereby developed was focused on the Software Developer Kit (SDK) for Kinect. The latter was complemented with a set of tools to facilitate the recognition and definition of various human gestures. The work was divided in two parts wherein the first consisted on the capture and storage of gestures. The latter are modeled via point mapping and stored in a library of pre-defined actions, like for instance sign language. On the second and last stage of the work, an algorithm is developed to aid the recognition of gestures in real-time by applying the matching user input with the available set of gestures previously added to the library.

Keywords:

SDK Extension

Gesture library

Recording gestures

Gesture real-time recognition

Kinect

Índice

1.	INTRODUÇÃO	1
1.1	Motivação.....	1
1.2	Objetivos.....	2
1.3	Contribuições Originais	3
1.4	Estrutura da Dissertação.....	3
2.	ESTADO DA ARTE.....	5
2.1	Natural User Interfaces - Onde é Usada	5
2.1.1	Jogos.....	6
2.1.2	Comunicação por Sinais	7
2.1.3	Nova Geração	8
2.2	Métodos de Reconhecimento	9
2.2.1	Reconhecimento Padrão	13
2.3	O Sensor Kinect	14
2.3.1	Introdução	14
2.3.2	Outros Dispositivos	15
2.3.3	Software Development Kit	18
2.4	Conclusão	19
3.	SOLUÇÃO PROPOSTA	21
3.1	Modelo de Funcionamento	21
3.2	Criação da Biblioteca de Gestos	25
3.2.1	Definição de Gesto	25
3.2.2	Gravação do Gesto.....	26
3.3	Ferramentas de Suporte.....	31
3.4	Reconhecimento em Tempo Real.....	35
3.5	Conclusão	38
4.	VALIDAÇÃO.....	41
4.1	Material e Métodos	41
4.2	Gestos.....	42
4.3	Resultados	45

4.3.1	Gravação do Gesto.....	45
4.3.2	Reconhecimento.....	47
5.	CONCLUSÕES E TRABALHOS FUTUROS	53
5.1	Conclusão	53
5.2	Trabalhos Futuros.....	54
	BIBLIOGRAFIA	55

Índice de Figuras

Figura 1.1. Significado de NUI.....	1
Figura 1.2. Sequência a seguir até ao reconhecimento do gesto.	3
Figura 2.1. Evolução da tecnologia. (Lieveld, 2013)	6
Figura 2.2. Evolução das televisões. (Burger, 2012)	6
Figura 2.3. Dispositivos de controlo de movimento (Wii, Move, kinect). (King, 2013) (Sony, 2013) (Techcyn, 2010)	7
Figura 2.4. Myo, um novo sistema de deteção de gestos. (Thornton, 2013)	8
Figura 2.5. WiSee, deteção de gestos por sinais wireless. (Sanders, 2013)	9
Figura 2.6. Algoritmo baseado em segmentação para descrever trajetórias.	11
Figura 2.7. Descrição de gesto através de conjunto de ângulos.....	12
Figura 2.8. Exemplo de Pattern Matching.	13
Figura 2.9. Algoritmo “Classes of characters”.	14
Figura 2.10. Hardware do Kinect Sensor. (Fernandez, 2012)	15
Figura 2.11. Luzes de infravermelhos. (Flatley, 2011)	15
Figura 2.12. Articulações detetadas pelo Kinect. (Raiten, 2011)	18
Figura 2.13. Classe DepthImagePoint.....	19
Figura 2.14. NUI na 7ª arte. (Noessel, 2013)	19
Figura 3.1. Diagrama de interação entre programador e a extensão do SDK.	21
Figura 3.2. Diagrama de funcionamento das equações.	22
Figura 3.3. Representação da classe gesture.	23
Figura 3.4. Representação da classe biblioteca de gestos.	24
Figura 3.5. Movimento da mão.	25
Figura 3.6. Segmento da equação.	25
Figura 3.7. Movimento com dois sentidos.	26
Figura 3.8. Definição de gesto por várias funções.....	26
Figura 3.9. Erro de captação.	27
Figura 3.10. Análise do threshold.	28
Figura 3.11. Trajetória limpa.	28
Figura 3.12. Repetição de gestos.	28
Figura 3.13. Redimensionamento da repetição de gestos.	29
Figura 3.14. Aplicação da média móvel simples.	30
Figura 3.15. Divisão de equações.....	31
Figura 3.16. Imagem capturada da camara de pré-visualização.	32
Figura 3.17. Pré-visualização das articulações através da camara de profundidade.	32
Figura 3.18. Janela de edição dos pontos capturados pelo Kinect.	33
Figura 3.19. Janela de visualização da equação em três dimensões.	34
Figura 3.20. Lista de gesto da biblioteca e histórico de reconhecimento.	35

Figura 3.21. Erro no reconhecimento.	36
Figura 3.22. Dimensão de tempo no reconhecimento de um gesto.	36
Figura 3.23. Remover pontos.	37
Figura 3.24. Representação do processo executado quando ativado o reconhecimento.	38
Figura 3.25. Diagrama do processo de gravação do gesto.	39
Figura 4.1. Disposição do cenário usado para teste.	41
Figura 4.2. Indicação dos módulos observados na validação.	42
Figura 4.3. Gestos simples utilizados no teste da aplicação. (Grupo I)	43
Figura 4.4. Gestos simples de várias articulações utilizados no teste da aplicação. (Grupo II)	43
Figura 4.5. Gestos complexos de várias articulações utilizados no teste da aplicação. (Grupo III)	44
Figura 4.6. Grupo de gestos com outros pontos do corpo. (Grupo IV)	44
Figura 4.7. Múltipla captação de pontos de 2 gestos diferentes.	45
Figura 4.8. Definição do gesto por equações utilizando a primeira captação de dados.	46
Figura 4.9. Definição do gesto por equações utilizando um filtro no conjunto das captações.	46
Figura 4.10. Taxa de sucesso de reconhecimento de gestos inseridos na biblioteca (Grupo III). ...	49
Figura 4.11. Taxa de reconhecimento, com equações criadas manualmente.	52

Índice de Tabelas

Tabela 2.1. Diferenças do Hardware.	16
Tabela 2.2. Diferenças de Software.	17
Tabela 2.3. Diferenças gerais.	17
Tabela 2.4. Pontos fracos dos projetos mencionados.	20
Tabela 4.1. Número de gestos reconhecidos do Grupo I.	47
Tabela 4.2. Número de gestos reconhecidos do Grupo II.	47
Tabela 4.3. Número de gestos reconhecidos do Grupo III.	48
Tabela 4.4. Taxa de sucesso para as equações criadas de forma manual.	51

1. Introdução

A dissertação enquadra-se na área das Interfaces Naturais. Neste primeiro capítulo é dado a conhecer alguma inovação neste tipo de interação e qual o problema que se pretende atacar. São ainda definidos os objetivos a alcançar sobre esse problema encontrado.

1.1 Motivação

As Interfaces Naturais, ou *Natural User Interfaces* (NUI), têm evoluído bastante nos últimos tempos, por forma a melhorar a comunicação homem-máquina. Esta interação é baseada essencialmente em gestos e comandos de voz. Para tal é necessário existir um dispositivo que possa captar todos estes sinais humanos e traduzi-los em comandos que a máquina possa entender. Os gestos são normalmente muito intuitivos, de forma a serem facilmente compreendidos por qualquer pessoa e em qualquer parte do mundo, e então associados a um determinado significado

«NUI não é uma **Natural User Interface**, mas mais uma interface que faz com que o ato do utilizador pareça natural. Uma maneira fácil de entender isto é mudar a maneira como se diz “*Natural User Interface*”. Não é **Natural User Interface**, mas sim **Natural User Interface**.» (Figura 1.1) (Wixon & Wigdor, 2011).

Not
NATURAL USER INTERFACE,
but
NATURAL USER INTERFACE

Figura 1.1. Significado de NUI.

Muitos dispositivos podemos encontrar facilmente no mercado, como o Kinect, Leap Motion e até Smart TVs. Qualquer um destes dispositivos consegue reconhecer um pequeno grupo de gestos humanos, criados singularmente. No caso do Kinect e Leap Motion são mesmo capazes de reconhecer algumas partes do esqueleto humano e fazer o seu mapeamento. No que diz respeito aos gestos reconhecidos pelos dispositivos, estes são ainda um número consideravelmente pequeno, logo existe uma necessidade de aumentar essa lista de gestos. O problema é que nenhum dos dispositivos permite a adição de novos gestos. Surge então a necessidade de construir uma estrutura que suporte a criação de novos gestos assim como a deteção dos mesmos.

Tendo como base o Kinect Sensor existem alguns problemas que têm de ser considerados à partida devido à qualidade do *hardware*, assim como a sensibilidade aos raios UV que nem sempre permitem devolver a informação correta sobre a localização do esqueleto humano. É necessário então uma estabilização do mapeamento para que possa captar de forma correta o movimento.

A captação do movimento é outro dos desafios visto que o Kinect apenas disponibiliza a localização atual de cada ponto do esqueleto e nunca posições anteriores, no sentido de compreender o gesto.

Pretende-se então a criação de uma base de conhecimento que inclua todos os gestos pretendidos numa primeira fase, e posteriormente é necessário o dispositivo ser capaz de identifica-los em tempo real, quando alguém fizer um gesto idêntico, numa segunda fase. Acresce, nesta segunda fase, a dificuldade de nunca se saber quando o utilizador pretende começar e acabar o gesto.

1.2 Objetivos

De um modo geral o objetivo desta dissertação é adicionar uma 4ª dimensão, o tempo, ao atual Software Development Kit (SDK) do Kinect. Para tal, será criada uma extensão do SDK que permita o reconhecimento de gestos em tempo real, tentando resolver os problemas anteriormente descritos.

A solução pretendida é a criação de duas novas estruturas:

- Criação de bibliotecas de novos gestos.

Esta primeira estrutura será responsável pela captação de gestos. Aproveitando as capacidades do Kinect, é estudado o gesto humano, sendo este inserido numa biblioteca criada pelo utilizador.

Uma biblioteca criada será um conjunto de gestos que o utilizador ache que façam sentido agrupar para o fim pretendido, como por exemplo um conjunto de gestos de língua gestual portuguesa, ou um conjunto de gestos para controlo de uma célula fabril.

- Deteção de gestos em tempo real.

Nesta segunda parte o objetivo a alcançar é o reconhecimento dos gestos, em tempo real. Por outras palavras o gesto deverá ser reconhecido assim que acabado de executar e sem qualquer indicação adicional de quando começou e quando terminou.

Estas duas novas ferramentas complementares pretendem ser uma mais-valia para o programador na simplificação da criação e deteção de gestos, a fim de integrarem diversas aplicações.

À semelhança do que acontece no SDK do Kinect, pretende-se facilitar as tarefas do programador, fornecendo um conjunto de ferramentas úteis à criação de gestos. O processo até ao reconhecimento do gesto pretende ser tão simples como mostrado na Figura 1.2: o utilizador executa o gesto pretendido enquanto o Kinect capta esse mesmo movimento, o gesto é enviado para a biblioteca e assim que o utilizador inserir todos os gestos pretendidos, basta ativar o reconhecimento e todos os gestos estarão habilitados ao reconhecimento.

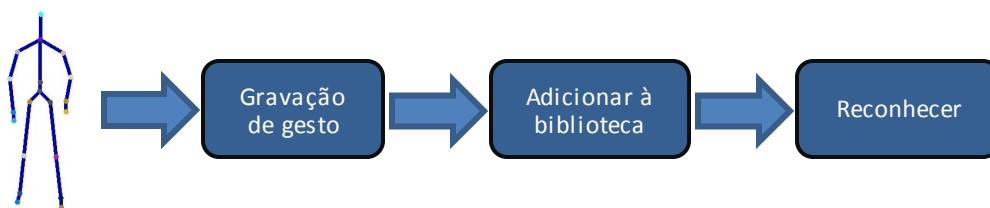


Figura 1.2. Sequência a seguir até ao reconhecimento do gesto.

1.3 Contribuições Originais

A contribuição desta dissertação é um modelo de criação e reconhecimento de novos gestos para sensores de movimentos. Este modelo, com o fim de integrar diversas aplicações, utiliza um reconhecimento em tempo real de forma a reconhecer gestos ao mesmo tempo que recebe informação do sensor.

Esta dissertação originou um artigo, submetido na revista *Creative Technologies* da European Alliance for Innovation.

1.4 Estrutura da Dissertação

Após esta introdução, este documento está organizado da seguinte forma:

- Capítulo 2: Estado da arte
É mostrado o que se tem feito no âmbito do reconhecimento de gestos e que propostas têm aparecido, as suas soluções e quais os seus principais problemas.
- Capítulo 3: Solução proposta
Este capítulo explica a solução proposta para os problemas mencionados no capítulo anterior. É feita uma abordagem aos principais algoritmos criados para a biblioteca de gestos, e explicado o seu funcionamento. Serão ainda mostradas algumas ferramentas de auxílio para criação e deteção dos gestos.
- Capítulo 4: Validação
São apresentados os métodos utilizados para a validação, a fiabilidade da solução proposta, as falhas existentes e quais as condições favoráveis para um melhor funcionamento.
- Capítulo 5: Conclusões e trabalhos futuros
Neste último capítulo é feita uma reflexão sobre o trabalho desenvolvido, quais os pontos fracos e o que ele vem resolver, quais os cenários em que pode ser aplicado e a indicação de alguns pontos futuros.

2. Estado da Arte

Neste capítulo aborda-se a área das *Natural User Interfaces*, como é usado e em que dispositivos podemos encontrar. São apresentados alguns métodos de definição de movimentos e de reconhecimento.

É também abordado o dispositivo que se vai utilizar, tanto numa forma de comparação com outros dispositivos como numa descrição mais específica do seu funcionamento.

2.1 Natural User Interfaces - Onde é Usada

Natural User Interfaces (NUI) é uma forma cada vez mais utilizada para estabelecer a comunicação com os dispositivos. Os atuais teclados e ratos tentam ser substituídos por métodos mais intuitivos. No entanto, estes dispositivos seguem também uma tendência de evolução no suporte de gestos.

Atualmente o controlo de um computador sem um rato e um teclado, para alguns, é ainda um pouco estranho. No caso da Microsoft, os seus sistemas operativos começam já a estar preparados para o fim destes dispositivos (Microsoft, 2013) através do suporte de gestos tácteis e reconhecimento de voz. Estas formas de controlo não são ainda totalmente apelativas. Enquanto isso os antigos dispositivos seguem alguma evolução no suporte de gestos. No caso dos ratos, existem já alguns modelos com suporte de gestos. A marca Apple, por exemplo, tem o seu modelo Magic Mouse (Apple, 2013) que possui uma superfície táctil com suporte de gestos. Os gestos suportados são simples toques na superfície e o deslizar de dedos, um ou dois e em qualquer um dos sentidos à semelhança do que acontece no *touchpad* dos portáteis.

Atualmente vive-se uma nova era nas *natural user interfaces*. Com o aparecimento do Kinect Sensor muitos outros dispositivos de captação 3D começaram também a surgir, a nível comercial. Anteriormente as maiores inovações surgiam ao nível do reconhecimento de voz. Como representado na Figura 2.1 cada vez mais importa a portabilidade e o comodismo, com o aparecimento destes dispositivos de captação 3D surge ainda uma maior liberdade na comunicação entre a máquina e o humano.



Figura 2.1. Evolução da tecnologia. (Lieveld, 2013)

Uma das formas mais comuns de NUI que encontramos é através de ecrãs tácteis, atualmente muito usados no controlo de *smartphones*, *tablets* e muitos outros dispositivos. No caso dos *smartphones* já existe controlo à distância por gestos. Esta funcionalidade é apenas ainda comercializada pela Samsung e requer um sensor de infravermelhos como suporte à tecnologia (Samsung, 2013).

Esta área de interfaces naturais está muito ligada ao comodismo e à simplicidade. Alguns dos dispositivos presentes nas nossas casas tendem a libertar-se do *hardware* de controlo à distância. No caso das televisões o controlo começa-se já a fazer à distância (Figura 2.2): o tipo Smart TV em algumas marcas, fornece comandos com suporte de alguma interface natural e especificamente na marca Samsung existe já controlo totalmente livre de *hardware* (Samsung, 2013).

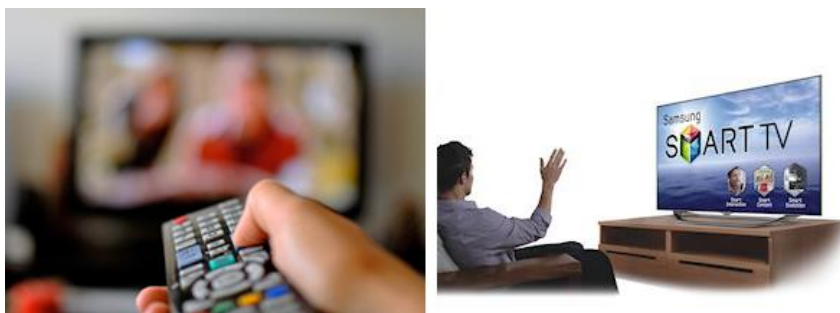


Figura 2.2. Evolução das televisões. (Burger, 2012)

2.1.1 Jogos

Atualmente são várias as áreas que integram esta interação usando novos sistemas de captação de movimento, como o Kinect Sensor.

A área dos jogos foi a que mais impulsionou estes sistemas de captação de movimento (Figura 2.3), o Kinect foi o primeiro sensor de movimentos a usar apenas câmaras para deteção de movimentos,

libertando o utilizador de qualquer tipo de comando físico. A consola de jogos Wii foi a primeira a “brincar” com o movimento humano. A captação deste movimento é feita através de vários sensores integrados no comando. Este sistema veio revolucionar interação com os jogos com uma interação mais física. O Kinect faz apenas uma monitorização à distância através das suas câmaras incorporadas para captar o movimento humano. Quanto ao sistema da marca Playstation (Move), pode-se considerar uma mistura entre os outros dois sistemas, um comando com vários sensores que ajuda, a câmara, na identificação do utilizador. Este sistema é aquele que melhor precisão tem.



Figura 2.3. Dispositivos de controlo de movimento (Wii, Move, kinect). (King, 2013) (Sony, 2013) (Techcyn, 2010)

De um certo modo pode-se considerar as anteriores formas de interação como formas de *input*, jogar através dos nossos próprios gestos. Por outro lado, há outra forma de utilizar os gestos. Muitos dos jogos mais conhecidos utilizam movimento humano na personagem do jogo, o que torna o jogo mais real, para isto é necessário usar sistemas de captação de movimento mais profissionais, mais precisos. Um dos exemplos é o Xsens MVN. Este sistema consiste em usar vários mini sensores no corpo humano que possibilitam uma análise em tempo real do seu movimento, num computador (Roetenberg, Luinge, & Slycke, 2009). A EA Sports é uma das grandes empresas que já experimentaram este sistema. Neste caso, usaram um jogador de futebol profissional para captar os movimentos do jogador, de forma a estudar as suas tendências e assimilar mais conhecimento na área (ESPN, 2010). Estes estudos e movimentos captados foram mais tarde replicados no jogo FIFA.

2.1.2 Comunicação por Sinais

À semelhança da comunicação verbal, existe também comunicação por gestos, por vezes muito mais universal. A maior desvantagem destas linguagens gestuais é o número reduzido de pessoas que as dominam.

Desde o aparecimento destes sistemas de captação de movimento mais comerciais, começaram a surgir novas ferramentas que visam interpretar estes gestos comunicativos. A lingual gestual é a forma de comunicação entre a comunidade surda/muda e é o exemplo mais comum que podemos encontrar.

Na China foi desenvolvido um projeto com o apoio da Microsoft Research Asia que pretende facilitar a comunicação entre as pessoas. Este projeto é basicamente um interpretador de gestos e funciona em dois sentidos. Num primeiro sentido funciona como tradutor de gestos, onde uma pessoa executa um gesto de língua gestual chinesa e ele devolve a palavra correspondente e num segundo sentido, onde o utilizador pode comunicar sem conhecer a língua gestual, neste caso pode-se escrever por texto e um avatar vai representar o gesto correspondente (Zhou, et al., 2013). É de referir que este projeto é suportado pelo Kinect.

Existem muitos outros tipos de linguagem gestual, como por exemplo, a comunicação que é feita entre um piloto de aviões e o sinalizador. Um dos projetos do Massachusetts Institute of Technology (MIT) tem como objetivo reconhecer um conjunto de gestos que é normalmente usado para comunicar entre a tripulação. Este projeto reconhece gestos contínuos tanto de alguns pontos do esqueleto como das mãos, sendo o grande desafio tentar que as aeronaves possam reconhecer as ordens sem margem de erro (Hardesty, 2012) (Song, Demirdjian, & Davis, 2012).

2.1.3 Nova Geração

Myo é um novo projeto na área do reconhecimento de gestos (Thalmic Labs, 2013). Este sistema diferencia-se dos outros por não usar nenhum dos sensores habituais. O Myo, como mostra a Figura 2.4, usa um sensor que deteta atividade muscular. Uma das vantagens ao usar este tipo de sensor é a liberdade na escolha do meio (local) onde se encontra, não dependendo de condições ideais.

O sistema, tal como muitos já existentes, é muito dirigido a programadores e à exploração de novas utilidades para diversas áreas.

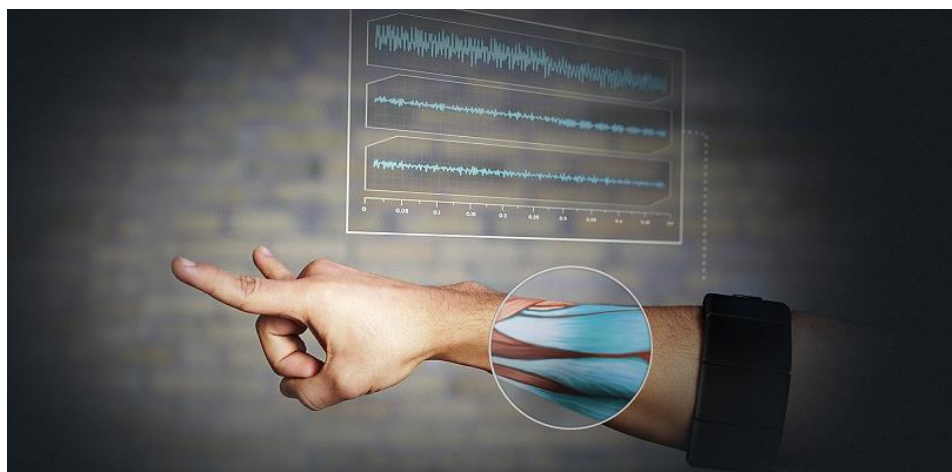


Figura 2.4. Myo, um novo sistema de deteção de gestos. (Thornton, 2013)

Outro sistema que promete vir revolucionar o controlo por gestos, desta vez sem sistema físico, é o WiSee (Pu Q. , Gupta, Gollakota, & Patel, 2013). Este sistema é completamente diferente de todos os outros por não possuir nenhum dispositivo físico, apenas utiliza sinais de *wireless* existentes.

Este projeto, desenvolvido na universidade de Washington, estuda os sinais de *wireless* (Figura 2.5) de forma a detetar variações e associando certas variações ao movimento humano. O projeto permite até ao momento a deteção de um conjunto de nove gestos com uma precisão de 94% (Pu Q. , Gupta, Gollakota, & Patel, 2013).

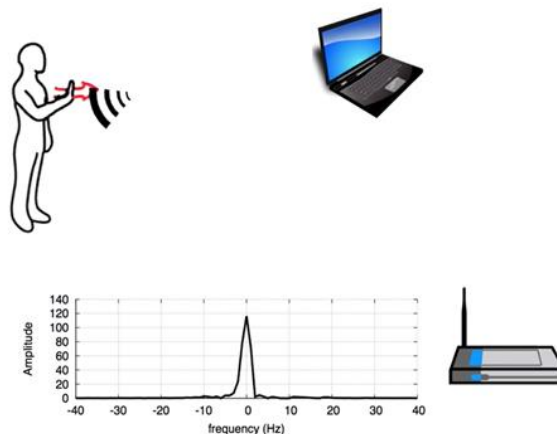


Figura 2.5. WiSee, deteção de gestos por sinais wireless. (Sanders, 2013)

2.2 Métodos de Reconhecimento

Na tentativa de aplicar os gestos como forma de controlo, surgem várias propostas e aplicações para uma criação dinâmica de gestos e também para a sua deteção. Estas aplicações são sempre compostas por duas fases, a primeira fase é sempre a criação do gesto e a segunda o seu reconhecimento.

Um dos principais focos é a definição do gesto, a forma como será representado, ou seja, as variáveis que serão definidas para descrever o gesto. Existem ainda outros dois pontos de foco, a maneira como é captado o gesto de forma a obter as variáveis necessárias e como é reconhecido o gesto, numa comparação com os gestos previamente definidos e armazenados.

O projeto **Gesture Recognition with a Wii Controller** (Schlomer, Benjamin, Henze, & Susanne, 2008) teve como objetivo criar um novo tipo de *input* à semelhança do rato, mas desta vez com o controlo *Wiimote* de forma mais livre, visto que é um dispositivo sem fios. Este controlo é o dispositivo que permite controlar a consola de jogos Wii da Nintendo. Internamente é composto por vários sensores que permitem detetar os seus movimentos.

Este projeto permitiu por parte do utilizador treinar alguns gestos que mais tarde podem ser usados para controlar vários sistemas, como por exemplo, o controlo de uma aplicação multimédia.

O método de definição dos gestos neste projeto consiste na representação de vetores tridimensionais. Estes vetores representam a aceleração captada pelo comando remoto, cada gesto pode ser treinado mais que uma vez de forma a obter melhor definição. A análise do gesto é feita através de reconhecimento de padrões do gesto, posteriormente diferenciados por uma taxa de comparação.

Como ponto negativo deste projeto pode-se referir a necessidade de usar um comando remoto, significando que apenas são possíveis movimentos efetuados pela mão que o segura. Isto é ideal para controlar aplicações de navegação, por exemplo.

No caso do projeto **Kinect SDK Dynamic Time Warping (DTW) Gesture Recognition** é usado um outro tipo de captação de dados, o Kinect. Este é um projeto *open source* e a finalidade é que o utilizador possa criar os seus próprios movimentos para um dado gesto.

Para efetuar a gravação, assim que é pressionado o botão de gravação o utilizador deve começar a realizar o movimento e este é gravado durante os 32 *frames* seguintes. O utilizador deve temporizar o gesto para que comece no primeiro *frame* e acabe no trigésimo segundo. Esta captação de movimento consiste apenas na posição, relativa ao centro do corpo, das articulações em todos os *frames*.

Como no caso deste projeto e em muitos outros apenas os membros superiores e por vezes apenas as mãos, são tidos em conta para a realização de um gesto. O desenvolvimento de projetos como estes, de reconhecimento de gestos, é muitas vezes pensado apenas para a navegação em aplicações.

Numa outra escala, o reconhecimento pode ser usado para interpretar qualquer tipo de gestos, considerando toda a estrutura do corpo. A sociedade tem tendência a usar alguns gestos na sua comunicação, havendo mesmo pessoas que utilizam apenas gestos para comunicar. O caso do **The Social Signal Interpretation (SSI) Framework** (Wagner, et al., 2013) é pensado para reconhecer qualquer gesto presente na sociedade. No que diz respeito à criação de gestos, esta *framework* usa linguagem XML para a descrição de cada gesto e tenta descreve-los da forma mais genérica possível. Variáveis como a duração mínima e máxima que são usadas para definir o tempo das ações, ponto de referência, posição da referência e outras, são usadas para diferenciar os gestos. Já o reconhecimento dos gestos é procurado dentro de um *buffer*, limitando o tamanho do gesto a este *buffer*.

Tendo como referência o artigo **Social Signal Interpretation (SSI) Framework**, o projeto **Full Body Interaction Framework** é um dos mais completos e de código aberto, na criação e deteção de movimentos referentes a todo o esqueleto humano. Existe uma diferenciação entre gestos estáticos e gestos contínuos. Nos gestos estáticos são guardadas as posições de cada uma das articulações e o objetivo é o utilizador igualar o esqueleto, por comparação de posição em cada uma das articulações. No caso dos gestos contínuos, a definição do gesto é composta por alguns pontos estáticos e a verificação é feita por etapas, isto é, para reconhecer o gesto é necessário que cada um dos pontos passe numa ordem correta em certas posições intermedias. É também possível uma combinação entre gestos estáticos e contínuos.

Estes gestos são descritos por posições mínimas e máximas, tempos de execução, vetores direcionais, ângulos e algumas outras variáveis. O seguinte código é um exemplo de como é representado o gesto do Zorro (Desenho da letra Z).

```
<LinearMovementRecognizer name="rightHandRight" visibility="hidden">
  <Joints main="rightHand" relative="rightShoulder"/>
  <Direction x="1" y="0" z="0" maxAngleDifference="30"/>
  <Speed min="1200"/>
</LinearMovementRecognizer>

<LinearMovementRecognizer name="rightHandLeftDown" visibility="hidden">
  <Joints main="rightHand" relative="rightShoulder"/>
  <Direction x="-1" y="-1" z="0" maxAngleDifference="30"/>
  <Speed min="1200"/>
</LinearMovementRecognizer>
```

(Kistler, 2011)

Este código representa dois movimentos lineares necessários para a composição do gesto referido, ambos referentes à mão direita, o ponto do esqueleto humano que irá realizar o gesto. Os dois gestos referem-se a duas translações da mão em sentidos diferentes e ambos com um tempo máximo de execução de 1200 milésimos de segundo.

A adição de gestos é efetuada através da criação de código semelhante, para um novo gesto é necessário que o programador entenda a linguagem, de formato XML, que define os gestos. Este *software* não dispõe de ferramentas de apoio à criação de gestos.

Um outro algoritmo interessante de observar encontra-se associado a mapeamento de objetos (Chen & Chang, 2000). Esta proposta tem como objetivo captar a trajetória de um objeto em vídeo de forma a poder comparar com outras trajetórias armazenadas numa base de dados.

O algoritmo de captação da trajetória e de reconhecimento é muito baseado em segmentos. Na captação, a posição de um objeto é adquirida e memorizada ao longo do tempo, como mostrado na Figura 2.6. A trajetória adquirida passa por vários passos sendo o primeiro uma suavização para uma melhor performance no tratamento de dados. O passo seguinte é a ideia principal em que este algoritmo é baseado, esta trajetória já previamente suavizada é dividida em pequenos segmentos para evitar trajetórias complexas. Estas pequenas segmentações são analisadas assumindo uma aceleração constante. Posteriormente estas informações como a aceleração, velocidade, comprimento de arco e outras, são adicionadas a cada segmento.

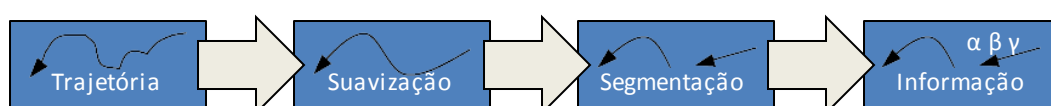


Figura 2.6. Algoritmo baseado em segmentação para descrever trajetórias.

Após esta definição da trajetória é efetuada uma pesquisa na base de dados para tentar comparar a trajetória adquirida. Cada segmento do movimento é comparado com alguns segmentos já conhecidos e memorizados na base de dados, assim o gesto será um conjunto de segmentos já conhecidos associados a algumas variáveis que o permitem diferenciar do segmento genérico.

No artigo **Handwritten Gesture Recognition for Gesture Keyboard** (Balaji, Deepu, Madhvanath, & Prabhakaran, 2006), a proposta consiste no reconhecimento de alguns caracteres. A captação de cada carater é obtida através do mapeamento do objeto que é usado para desenhar o carater, ou seja, é usado o movimento efetuado para criar cada carater na sua identificação.

Um dos métodos usados para caracterizar um movimento, interpretado como uma linha, é a sua descrição através de várias características globais. A caracterização usada é o comprimento total da linha, a razão entre as duas extremidades e o seu comprimento total, a orientação que as duas extremidades fazem com a horizontal, uma fração dos pontos na metade superior da área e uma soma entre todos os ângulos de pequenos segmentos numa aproximação á linha.

De salientar que neste artigo existe um número limite de gestos a ser reconhecido, o número de caracteres do alfabeto Devanagari, foi por isso possível arranjar um conjunto de 13 segmentos em que se pode decompor todos os gestos.

O artigo **Human Hand Detection for Gestures Recognition of A Partner Robot** (Kubota & Abe, 2006) propõe uma comunicação com um robô através de gestos. É interessante observar uma definição do gesto através de uma composição de ângulos. A Figura 2.7 mostra que várias amostras, captadas em cada *frame*, são guardadas sequencialmente com o angulo de cada articulação para a definição do gesto.

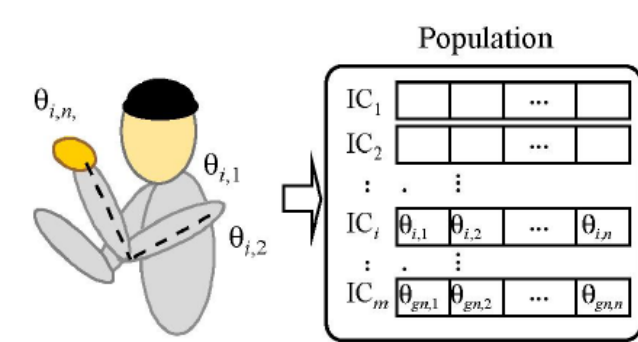


Figura 2.7. Descrição de gesto através de conjunto de ângulos.

Na área da saúde, um dos projetos existentes (Nomm, Leibak, Vain, & Miyawaki, 2010) permite, a robôs conhecidos como *scrub nurse robot*, a análise ao movimento da mão do cirurgião e reconhecimento de alguns gestos, comparando com outros gestos humanos. Esta análise e reconhecimento é baseado em pequenas comparações, o movimento é dividido em pequenas áreas. O conjunto destas áreas será a composição de um gesto, no que diz respeito ao reconhecimento, ao longo da representação do

gesto as áreas são preenchidas com o movimento, que é comparado com um movimento humano previamente guardado.

De um modo geral parece haver alguma convergência na definição do movimento de um gesto, num modo discreto os movimentos são descritos com um conjunto de vários pontos (x, y) que seguem uma certa orientação e num modo contínuo onde por vezes são apresentados como um trajeto de segmentos retos e outras vezes por um conjunto de outros segmentos conhecidos.

2.2.1 Reconhecimento Padrão

Pattern Matching é uma técnica muito usada no que diz respeito a algoritmos de reconhecimento. Este conceito, de um modo geral, tem como objetivo reconhecer padrões no meio de informação. Um exemplo fácil de compreender é encontrar um conjunto de caracteres no meio de outros tantos (Figura 2.8), o exemplo que se segue mostra como encontrar a sequência “AABA” entre um conjunto de caracteres, procurando-a passo a passo (Sedgewick, 2007).

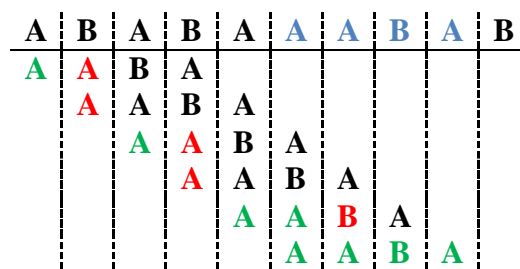


Figura 2.8. Exemplo de Pattern Matching.

Numa outra perspetiva, o exemplo anterior facilmente pode ser interpretado como símbolos e não apenas caracteres, trata-se de procurar um conjunto de símbolos ordenados em um conjunto aleatório de vários outros símbolos.

Gonzalo Navarro refere numa das suas publicações (Navarro, 2004) alguns algoritmos que são aplicados na área de reconhecimento de padrões, desde o mais simples, como o exemplo anterior, ao mais complexo, como alguns dos algoritmos que se seguem.

Um dos algoritmos referidos é muito similar ao acima mostrado, mas desta vez não se pretende reconhecer uma sequência fixa de caracteres mas sim uma sequência variável. O Algoritmo representado na Figura 2.9 pretende reconhecer um conjunto de 3 caracteres, em que o primeiro é ‘A’ ou ‘B’, o segundo ‘C’ ou ‘D’ e o último é o carácter ‘E’. Como representado, são seguidos 3 passos até o padrão ser reconhecido.

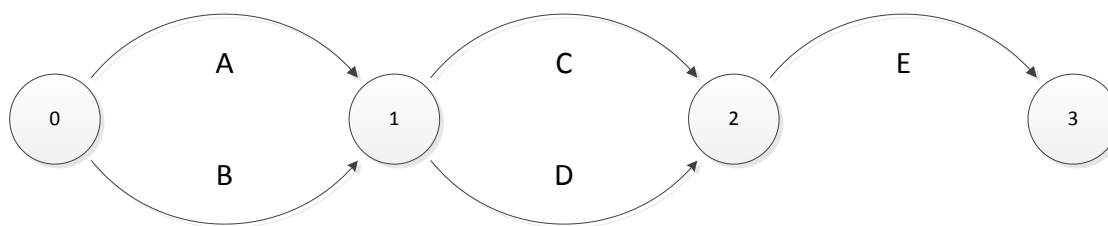


Figura 2.9. Algoritmo “Classes of characters”.

Estas pequenas demonstrações de algoritmos mostram que é possível reconhecer padrões no meio de vária informação, apenas temos de construir o algoritmo certo para o padrão pretendido. Estes padrões não são aplicados apenas a texto, qualquer tipo de variável é válida. O meio a que se pretende aplicar este tipo de reconhecimento deve ser com as variáveis adequadas, para então se definir um padrão com algumas das variáveis.

2.3 O Sensor Kinect

2.3.1 Introdução

Originalmente apresentado como “Projeto Natal”, o Kinect foi lançado no fim do ano de 2010 como acessório da consola de jogos Xbox 360, da Microsoft. Este projecto foi desenvolvido em parceria com a empresa PrimeSense que mais tarde novamente com uma parceria, desta vez com a Asus, desenvolveu um sistema idêntico (Xtion) mas com o intuito de servir computadores. Foram também membros fundadores de um dos mais poderosos SDK para estes sistemas (OpenNI, 2013), o OpenNI, trata-se de um SDK *open source* inicialmente utilizado no Kinect.

Criado com o intuito de dar um novo dinamismo na área dos jogos, substituindo o tradicional comando para um controlo mais intuitivo, o Kinect rapidamente alargou a visão sobre as áreas de aplicação.

Em relação à sua estrutura física, o Kinect dispõe de uma câmara RGB, um emissor e recetor de infravermelhos (Depth Sensor), um conjunto de microfones e ainda um motor na base que permite mexer ligeiramente toda a estrutura (Figura 2.10).



Figura 2.10. Hardware do Kinect Sensor. (Fernandez, 2012)

A câmara RGB permite a captura de três canais até uma resolução de 1280x960, a junção dos três canais (vermelho, verde e azul) permite que a imagem seja colorida. Através desta câmara já é possível fazer muito processamento de imagem, no entanto é como utilizar uma câmara normal, a maior inovação do sistema vem na união com a câmara de profundidade.

A câmara de profundidade, como o próprio nome indica, serve para dar a noção de profundidade a que as coisas se situam. O emissor emite um feixe de luzes de infravermelhos e o sensor de profundidade lê essas luzes refletidas como apresenta a Figura 2.11.



Figura 2.11. Luzes de infravermelhos. (Flatley, 2011)

Um ponto importante a referir é que esta câmara é muito sensível aos raios UV, é aconselhável usar o Kinect numa sala protegida da luz solar. Outra das recomendações, para o bom funcionamento da versão do Kinect para consolas as especificações sugerem que o utilizador se deve situar a uma distância de pelo menos 1,80 metros.

2.3.2 Outros Dispositivos

Tendo como principal objetivo a criação de uma biblioteca de gestos existem vários sistemas alternativos ao escolhido. Alguns desses sistemas têm características que permitiriam um melhor

desempenho, no entanto têm também outras características desfavoráveis. Para esta comparação com o Kinect foram usados quatro dispositivos com características significativamente diferentes. O Xtion, já referido anteriormente, é um dispositivo muito semelhante ao Kinect. Já o Leap Motion distingue-se pela sua precisão e tamanho, a Smart TV é um sistema um pouco diferente de todos os outros por não ser tão móvel e não ser ligado a um computador, no entanto começa a ser muito utilizado e dispõe do seu próprio processamento e é sempre importante comparar com um sistema mais simples e acessível a grande parte das pessoas, neste caso, uma Webcam.

A Tabela 2.1 mostra uma comparação entre a parte física dos dispositivos. O Kinect e o Xtion distinguem-se pela diversidade e o Leap Motion pela dimensão.

Tabela 2.1. Diferenças do Hardware.

		Kinect	Xtion	Leap Motion	Smart TV	Webcam
Hardware	Camera RGB	Sim	Sim	Não	Sim	Sim
	Camera de profundidade	Sim	Sim	2	Não	-
	Dimensão	Média	Média	Pequena	Grande	Pequena

No que diz respeito ao *software* existente para cada um dos dispositivos, a Tabela 2.2 marca algumas das diferenças sobretudo no que diz respeito aos kits de desenvolvimento oficiais. No caso das Smart TV e Webcams não pertencem a nenhuma marca, logo a existência de SDKs oficiais é relativo, mas em geral todas as marcas de Smart TVs disponibilizam um SDK. A Smart TV testada na Tabela 2.2 refere-se à marca Samsung.

Os dispositivos Kinect e Leap Motion têm um grande apoio da marca no desenvolvimento, no caso do Xtion é utilizado o SDK *OpenNI* que pode ser utilizado também pelo Kinect. Em relação à quantidade de gestos conhecidos e detetados pelos sistemas é muito fraca, sendo o Leap Motion que mais se preocupa com essa área.

A Webcam tem como vantagem de ser muito utilizada e acessível favorece a criação de vários algoritmos por parte dos utilizadores. Surgem muitos projetos na área de processamento de imagem, entre os quais deteção de dedos e reconhecimento facial. É possível obter informação da localização de pontos específicos do corpo com alguma fiabilidade.

Tabela 2.2. Diferenças de Software.

		Kinect	Xtion	Leap Motion	Smart TV	Webcam
Software	SDK Oficial	Sim	Não	Sim	Sim	Não
	Pontos detetados	Sim	-	Sim	Não	-
	Tracking	Sim	-	Sim	Não	-
	Deteção facial	Sim	-	Não	Não	-
	Deteção de dedos	Não	-	Sim	Sim	-
	Quantidade de gestos reconhecidos	Baixa	-	Média	Baixa	-
	Quantidade de algoritmos não oficiais	Elevada	Elevada	Média	Baixa	Muito elevada

Na generalidade existem uns sistemas mais avançados que outros, a Tabela 2.3 mostra algumas diferenças gerais entre eles. No que diz respeito à comercialização, as *webcams* são os dispositivos a que mais pessoas têm acesso, as Smart TV são também bastante comercializadas, no entanto, o Kinect Sensor é o sistema mais usual no que diz respeito à captação de movimentos. Peca na qualidade de *hardware* disponibilizada o que não acontece nos anteriores e no *Leap Motion*.

Uma das grandes vantagens dos sistemas desenhados para a captação de movimento (Kinect, Xtion e Leap Motion) é noção de profundidade, no caso dos outros dispositivos é necessário um maior processamento de imagem para chegar à mesma informação. Por exemplo, enquanto nos dispositivos com noção de profundidade é fácil achar as articulações do corpo humano, nos outros é necessário analisar uma imagem completa, apenas com duas dimensões, onde se situa o corpo humano e a partir daí começar a analisar ponto por ponto.

Tabela 2.3. Diferenças gerais.

		Kinect	Xtion	Leap Motion	Smart TV	Webcam
Geral	Comercialização	Média	Baixa	Baixa	Elevada	Muito elevada
	Alcance	Longo	Longo	Baixo	Longo	Longo
	Qualidade	Baixa	Baixa	Alta	Média	Média
	Noção de profundidade	Sim	Sim	Sim	Não	Não

De um modo geral pode-se dizer que o Kinect Sensor é o dispositivo que melhor está preparado para fazer um reconhecimento de gestos, perde um pouco na qualidade do *hardware*, na precisão, o que

não acontece no Leap Motion. Este último apesar de ainda pouco usado é mais específico para trabalhar com mãos e pequenos objetos.

O Xtion é muito idêntico ao Kinect mas possui menos ajuda (SDK) por parte da marca e é muito menos popular.

Por fim numa comparação dos dispositivos mais comuns, Smart TV e Webcam, necessitam de uma grande quantidade de processamento para identificar o pretendido mas podem ser uteis para coisas mais simples. No caso das Webcams, por existir em grande número e em qualquer lugar se pode encontrar uma, existem tentativas interessantes na área do reconhecimento.

2.3.3 Software Development Kit

O SDK do Kinect disponibiliza várias ferramentas de apoio ao programador para a interação com o dispositivo físico (Microsoft, 2013).

Para a realização desta dissertação existem alguns aspetos importantes a referir sobre o funcionamento do SDK. O SDK está preparado para funcionar a várias frequências, sendo a frequência máxima de 30 frames por segundo (fps) e será esta a frequência que vai ser utilizada na biblioteca de gestos. Esta frequência significa que o processamento será executado 30 vezes num segundo, assim como o acesso à informação disponibilizada.

Uma das grandes utilidades do SDK é o reconhecimento de vários pontos do esqueleto humano. A versão 1.8 do SDK permite o reconhecimento de 20 articulações, em modo de esqueleto completo. Na Figura 2.12 estão representadas as articulações detetadas pelo sistema, qualquer uma das câmaras deteta e devolve a localização as articulações, mas só a câmara de profundidade devolve a localização com as 3 dimensões e com uma maior precisão, visto que é a única que tem capacidade para tal.

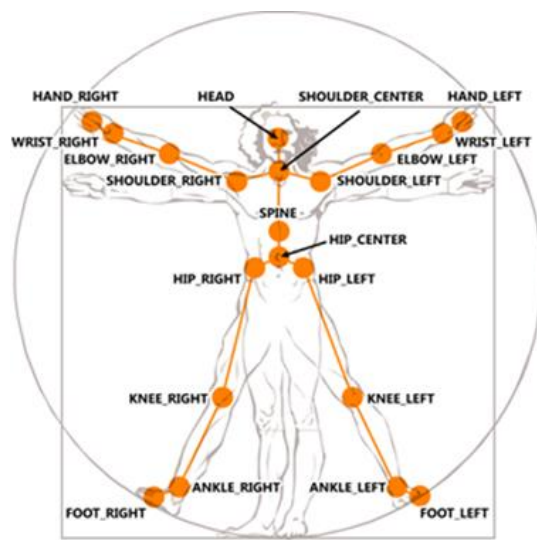


Figura 2.12. Articulações detetadas pelo Kinect. (Raiten, 2011)

Após a detecção do esqueleto humano o sistema de mapeamento toma conta de cada articulação, sendo esta a informação que vai ser disponibilizada ao utilizador/programador consoante a frequência solicitada.

A informação usada, do SDK para a construção da biblioteca é apenas informação básica. A Figura 2.13 mostra alguns dados que podem ser obtidos através do mapeamento das articulações. Na prática vão ser lidos os valores da variável *X*, *Y* e *Depth*. Estas variáveis surgem com valores novos à frequência de 30 fps.

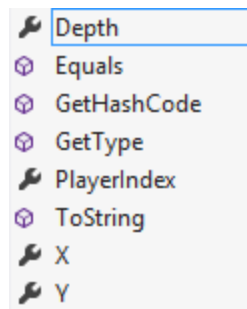


Figura 2.13. Classe DepthImagePoint.

Esta estrutura de dados surge para todas as articulações do esqueleto. Existe também uma lista com as 20 articulações referidas que simplifica o acesso à informação.

2.4 Conclusão

NUI é um modo de interação cada vez mais utilizado, existindo já uma grande tendência para a sua utilização nas novas tecnologias. Um dos grandes indicadores de como se imagina o futuro costuma ser a 7ª arte, como mostra a Figura 2.14. O controlo por gestos é já visto como um dos principais meios de comunicação com os dispositivos (Noessel, 2013).

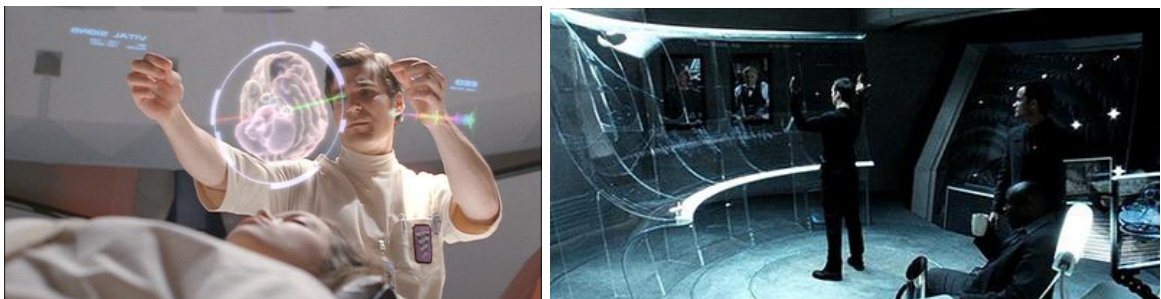


Figura 2.14. NUI na 7ª arte. (Noessel, 2013)

A área da diversão veio oferecer meios para que este modo de interação começasse a evoluir. Como por exemplo, o Kinect que começou a ser usado e olhado para outros fins que não os jogos. Com isto, começaram a surgir e a ser pensados outros sistemas fora da área dos jogos.

Entre jogos e outras áreas, muitas das aplicações que podemos encontrar, com reconhecimento de gestos, requerem os seus próprios gestos. Ao nível empresarial, o programador desenvolve apenas o reconhecimento para os gestos necessários, não tendo grande interesse em desenvolver um reconhecimento genérico. Existem por isso poucos projetos que visam a criação e deteção de gestos de forma genérica.

A Tabela 2.4 mostra alguns dos pontos fracos dos projetos já mencionados.

Tabela 2.4. Pontos fracos dos projetos mencionados.

Pontos fracos
<ul style="list-style-type: none">• Utilização de comandos físicos• Número reduzido de pontos do esqueleto• Definição do tempo de gravação• Reconhecimento em tempo real• Definição/Criação manual do gesto

Uma das características que esta dissertação tem em conta é a liberdade do utilizador. Será usado o sistema Kinect que permite ao utilizador comunicar sem o suporte de qualquer outro *hardware*. Este sistema permite também a deteção de vários pontos do esqueleto, enquanto alguns sistemas que necessitam do auxílio de um comando apenas reconhecem o movimento do próprio comando.

Existe alguma complexidade na construção do gesto, muitas vezes os gestos têm de ser definidos por código criando alguma dificuldade ao programado. No caso de o gesto poder ser gravado, a definição do tempo de gravação é um aspeto muito importante, é necessário definir bem os limites do gesto. Estes limites são também uma dificuldade no reconhecimento em tempo real, é difícil saber quando se deve tentar começar o reconhecimento do gesto e quando ele acaba, para que se possa fazer a comparação.

Estes vão ser os pontos a ter em conta na criação de uma biblioteca de reconhecimento de gestos.

3. Solução Proposta

No próximo capítulo é descrito o modelo de funcionamento para o problema encontrado.

Primeiro é feita uma abordagem à forma de como o movimento vai ser definido e guardado. Através desta forma de definição é explicado como funciona a captação e o reconhecimento de um gesto.

3.1 Modelo de Funcionamento

A proposta desta dissertação é a criação de uma nova camada de abstração de programação para os dispositivos de interfaces naturais. Com base nas características apresentadas no capítulo anterior, o dispositivo escolhido será o Kinect Sensor, por forma a melhor possibilitar a captação de gestos.

Para tal, genericamente, há que criar uma camada de abstração para os sistemas que utilizem interfaces naturais, em particular para o Kinect Sensor. A ideia é estender o SDK oficial que vem associado ao Kinect, por forma a acrescentar um novo conjunto de funcionalidades úteis ao programador, como esquematizado na Figura 3.1.

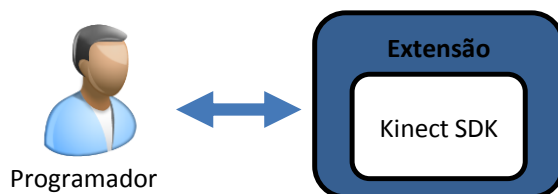


Figura 3.1. Diagrama de interação entre programador e a extensão do SDK.

Este conjunto de funcionalidades está dividido em dois tempos: modelação e criação de gestos e o seu reconhecimento em tempo real.

1. Modelação e criação de gestos – nesta fase, haverá que gravar um conjunto de gestos considerados para alimentar uma biblioteca. Os gestos serão os que o utilizador considerar necessários para o fim pretendido.
2. Reconhecimento – após a criação de uma biblioteca é feito o reconhecimento em tempo real de gestos. Assim que o sensor começa a captar movimentos e encontrar correspondência com um gesto dessa biblioteca, esse movimento é identificado.

Estas novas funcionalidades são pensadas em torno da modelação de gestos. Esta modelação consiste numa definição do gesto através de um conjunto de equações. A Figura 3.2 mostra a importância deste tipo de dados usado. Após uma gravação são obtidas equações e após as equações obtidas é possível reconhecer gestos.

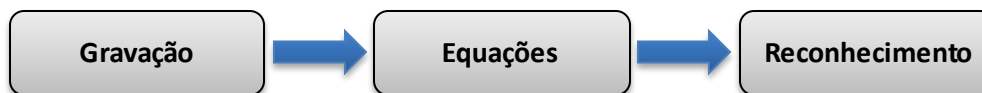


Figura 3.2. Diagrama de funcionamento das equações.

A primeira fase destina-se mais a uma pré-produção para o programador, algo que poderá ser criado previamente para depois ser utilizado na sua aplicação. Esta vai ser representada por uma classe (*Gesture*). A Figura 3.3 mostra uma visão geral do que é esta classe as suas principais características.

- Nome – O nome será um campo único para que se possa distinguir os vários gestos a adicionar na biblioteca.
- Matriz de posições – É essencial para a captação do movimento, a sua função é guardar as posições absolutas de cada articulação ao longo do tempo, para isso a matriz é diferenciada pelo índice de tempo que por sua vez engloba o índice de cada uma das articulações representadas no SDK. É dentro deste índice de articulações que será memorizada a posição de coordenada (x, y, z) captada.
- Ativo – Será uma pequena informação do estado da *class*, informa se está a decorrer alguma gravação do gesto em causa. Importante no sentido de anular algumas funções durante a sua gravação.
- Lista de articulações – Um dos espaços desta classe será uma lista com as várias articulações que poderão fazer parte ou não do gesto, cada elemento da lista é diferenciado pelo nome da articulação. Este espaço vai conter também várias gravações efetuadas ao gesto em questão.
 - Articulação ativa – Como já referido a articulação pode pertencer ou não ao gesto, sempre que ela seja parte do gesto este campo é ativado.
 - Gravações – Este será o campo que armazenará as várias gravações do gesto, o utilizador poderá adicionar uma gravação sempre que pretender, cada uma destas gravações será modelada e definida por alguns campos como o *Offset* e o *Resize*, mais à frente explicados.
 - Equações – As equações em questão serão duas representações matemáticas do gesto executado, uma referente ao plano (x, y) e outra ao plano (x, z) . Este campo poderá ser uma lista ou apenas um elemento, depende da complexidade do gesto.
 - Equações prontas – Um gesto apenas poderá ser adicionado à biblioteca se a transformação para as equações for efetuada com sucesso, porque é através delas que vamos trabalhar numa próxima fase. Se este campo estiver ativo então o gesto está pronto a ser adicionado.

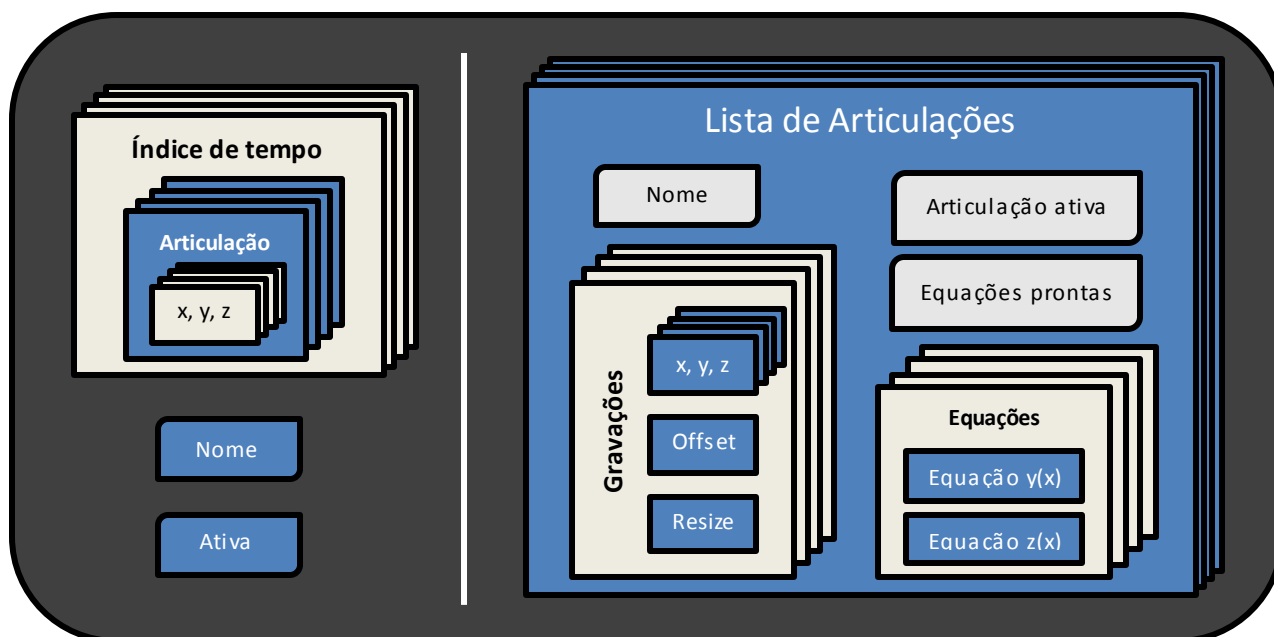


Figura 3.3. Representação da classe *gesture*.

As funcionalidades desejadas prendem-se com as operações a fazer para gerir a referida biblioteca de gestos, esta ultima *class*, apresentada incorporadas algumas funções uteis para a criação do gesto.

- Captação de movimento – esta função é responsável por captar e guardar todas as posições do esqueleto humano durante um certo período de tempo, fica disponível apenas em memória virtual.
- Verificar lista de articulações – uma função meramente informativa, disponibiliza ao utilizador quais as articulações a que o gesto é sensível.
- Adicionar gravações do movimento – pretende-se com esta função guardar mais que uma captação de movimento referente ao mesmo gesto, adicionar mais gravações significa uma melhor definição do gesto.
- Visualizar gesto em 2D – a fim de o utilizador ter uma maior noção do movimento representado, posteriormente a uma gravação pode visualizar as amostras (pontos) captadas do movimento de uma dada articulação.
- Visualizar gesto em 3D – após criação das equações a guardar na biblioteca é possível o utilizador confirmar e visualizar essas equações numa perspetiva a três dimensões.

A segunda fase corresponde à biblioteca de gestos, esta parte é essencial para o programador incluir na sua aplicação e a única que pode ser guardada fisicamente em memória. A Figura 3.4 mostra a visão geral de como será a classe (*GestureLibrary*) da biblioteca de gestos e o que a define.

- Nome – mera definição, dada pelo utilizador, como identificação da biblioteca de gestos.

- Detecção de movimento – esta é uma informação sobre o estado da biblioteca, sendo o objetivo principal detetar o gesto, este campo informa se a biblioteca se encontra em modo de deteção ou permanece inativada.
- Gestos – a biblioteca reserva o maior espaço para uma lista de gestos, estes serão os gestos previamente criados e adicionados que esta biblioteca irá detetar.
- Lista de pontos – à semelhança da matriz de posições, da classe de gestos, esta é uma lista de elementos indeterminados que pretende guardar todas as posições, de todas as articulações, a partir do momento a que a deteção de movimento for ativada.

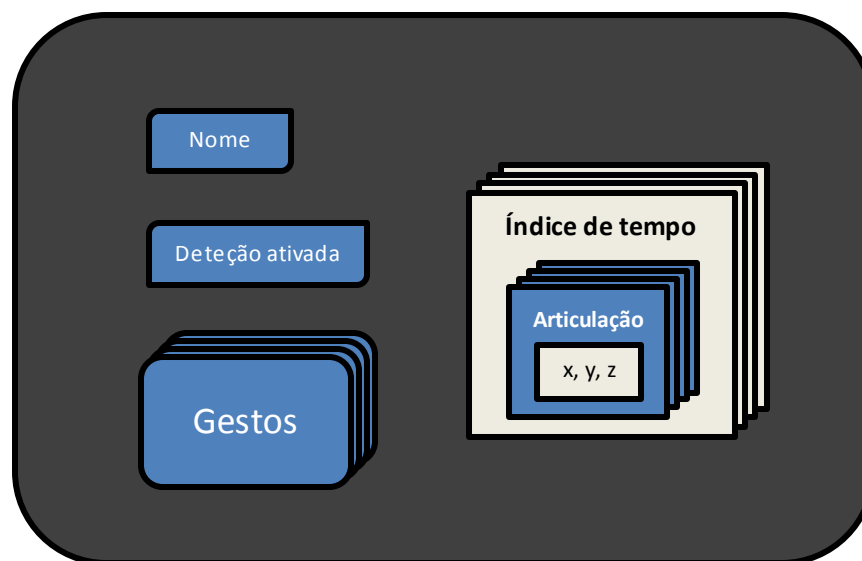


Figura 3.4. Representação da classe biblioteca de gestos.

Esta classe incorpora também algumas funções essenciais para o seu funcionamento.

- Ler / Gravar dados – esta função é simplesmente para carregar os gestos para a biblioteca ou guardar em disco.
- Adicionar / Remover gestos – permite adicionar, na biblioteca, gestos criados anteriormente na classe de gestos. É também possível excluir qualquer gesto.
- Verificar gesto – o utilizador pode verificar se um dado nome existe, na biblioteca, como gesto.
- Ver gestos – além da pesquisa por nome, o utilizador pode pedir os gestos incorporados nesta biblioteca e visualizar mais algumas características de cada gesto.
- Detetar – esta é a principal função e mais complexa da biblioteca de gestos, a sua função é captar todos os movimentos em tempo real e comparar com os gestos existentes na biblioteca.

3.2 Criação da Biblioteca de Gestos

3.2.1 Definição de Gestos

Para se poder armazenar um gesto numa biblioteca, há que, em primeiro lugar, decidir como é representado esse gesto.

Um gesto não é mais que um movimento do corpo, numa definição popular, os gestos servem para comunicar, exprimir ideias ou sentimentos. Mas no âmbito desta dissertação considera-se qualquer movimento um gesto, como por exemplo, um simples deslocamento da perna poderá ser considerado um passo.

Sabendo como funciona o SDK do Kinect, o foco principal será na posição de cada articulação do corpo. Para se captar o gesto, primeiro é preciso saber quais as articulações que queremos considerar, só essas vão representar o gesto. Como na Figura 3.5 a articulação, neste caso a mão, sofre um deslocamento e será esse deslocamento associado, há articulação que queremos gravar.

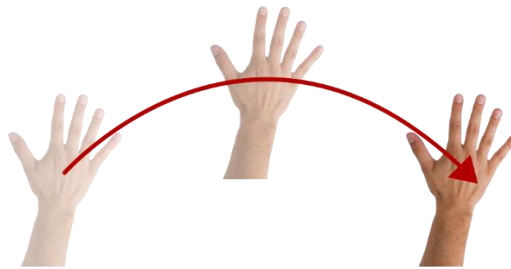


Figura 3.5. Movimento da mão.

De forma a guardar o movimento na biblioteca de gestos, este movimento é traduzido em equações polinomiais de grau 3. O movimento da Figura 3.5 é um dos exemplos mais simples que se pode ter, para tal, é preciso identificar a equação e o intervalo de tempo em que podemos encontrar uma aproximação deste segmento, como à semelhança da equação (3.1). Nenhum intervalo escolhido da uma equação polinomial pode descrever o gesto na perfeição. Porém, o segmento da equação da Figura 3.6 no intervalo $[a, b]$ pode ser muito idêntico ao movimento da Figura 3.5.

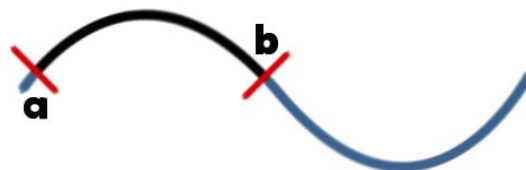


Figura 3.6. Segmento da equação.

$$y = a_3x^3 + a_2x^2 + a_1x + a_0 \in [a, b] \quad (3.1)$$

Apesar do movimento humano ser de três dimensões, será sempre definido por duas equações de duas variáveis e não uma equação de três variáveis, de forma a garantir equações mais simples. De acordo com o sistema de coordenadas do SDK, haverá uma equação para a coordenada y e outra para a coordenada z , ambas em função de x .

Um dos primeiros problemas da definição do movimento é o sentido do movimento, não é possível criar uma equação polinomial com dois sentidos como na Figura 3.7, neste caso o movimento terá de ser definido como um conjunto de equações.



Figura 3.7. Movimento com dois sentidos.

No caso de movimentos de maior complexidade, será difícil representar em apenas uma simples equação. À semelhança de quando o movimento inverte o sentido, a definição de um gesto complexo será descrita por n equações. A Figura 3.8 mostra a ideia de como um gesto mais complexo pode ser composto por várias equações.

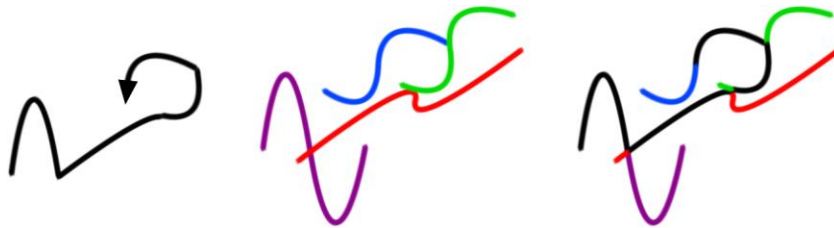


Figura 3.8. Definição de gesto por várias funções.

3.2.2 Gravação do Gesto

O processo de gravação do gesto vai desde a pré-gravação, definição de alguns atributos do gesto, até à pós-gravação, transformação do movimento e equações. Este processo pode ser descrito em três fases.

- Fase de captação – Nesta fase são definidos alguns atributos do gesto e de seguida feito a captação do movimento, com o Kinect, consoante os atributos definidos. Aqui pode ser feita mais do que uma captação do mesmo gesto para que seja mais preciso.

- Fase de modelação – A fase de modelação diz respeito às várias gravações que foram efetuadas para o mesmo gesto. Existe a possibilidade de editar cada uma das gravações e posteriormente redimensionar para que coincidam minimamente.
- Fase de transformação de dados – Após os gestos estarem perfeitamente modelados, será feita uma análise ao movimento e descrito em equações, de forma a diminuir a quantidade de dados e otimizar o processo de reconhecimento em tempo real.

Captação de movimentos

Na pré-gravação é necessário definir algumas coisas, como o tempo que necessitamos para gravar o gesto e quais as articulações que pertencem ao gesto. Quanto ao tempo será uma mera definição em segundo que o dispositivo (Kinect) estará sensível à captação dos pontos do esqueleto humano, já as articulações pertencentes será um campo, da classe de gestos, necessário a preencher com uma lista de articulações enumeradas no SDK.

Considerando a frequência de captação utilizada, 30 fps (*frames* por segundo), o número de pontos analisados de uma articulação é dado pela equação (3.2).

$$\text{número de pontos} = \text{número de segundos} \times 30 \quad (3.2)$$

Após a captação de todos estes pontos, é necessário fazer uma análise conseqüente da má precisão do dispositivo. Como na Figura 3.9, por vezes o Kinect fornece informação errada sobre a posição. Na tentativa de colmatar este erro é aplicado um algoritmo que visa melhor a precisão.

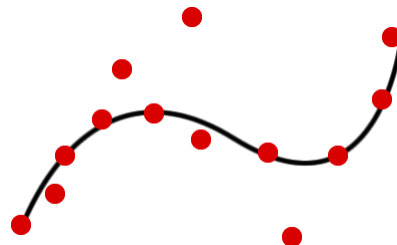


Figura 3.9. Erro de captação.

Este algoritmo tenta também reduzir o excesso de informação. A análise é sempre feita ponto a ponto e através da posição relativa ao último ponto. Especificamente, é definido um mínimo e um máximo, um *threshold*, e só será aceite uma posição seguinte que se enquadre dentro desse *threshold*. Assumindo que na Figura 3.10 a evolução dos pontos segue da esquerda para a direita e considerando o raio verde a distância mínima a que um ponto deve estar, para ao haver excesso de informação, e o raio vermelho a distância máxima a que um ponto deve estar. No caso da Figura 3.10 o ponto a vermelho será excluído sendo o ponto verde o próximo ponto da trajetória.

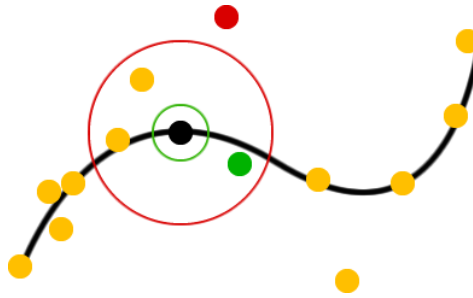


Figura 3.10. Análise do threshold.

Na distância máxima do próximo ponto é tido em conta uma normal velocidade do movimento humano, esta distância é definida apenas para garantir que não surjam posições impossíveis num movimento relativamente rápido. Após a aplicação deste algoritmo é possível obter resultados como o da Figura 3.11.

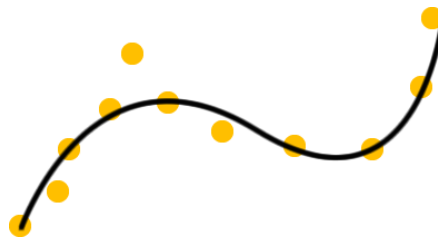


Figura 3.11. Trajetória limpa.

Modelação do movimento

Assim como na aplicação do algoritmo do *threshold*, esta fase de modelação visa também melhorar a precisão da trajetória, desta vez através da repetição, quantas vezes desejar, do gesto.

Durante a gravação do gesto pode haver algumas partes que queremos excluir, como o início ou o fim do gesto. O tempo definido para a gravação é o tempo em que temos a certeza que conseguimos representar o gesto e não o tempo exato que o gesto demora, por isso, umas das funcionalidades na criação do gesto é a redefinição de onde o gesto acaba ou onde começa.

Outra das inconveniências que encontramos é o não enquadramento do gesto repetido, como é mostrado na Figura 3.12, a repetição de um gesto nem sempre coincide com o anterior, o tamanho e a posição dos pontos analisados normalmente não são coincidentes.

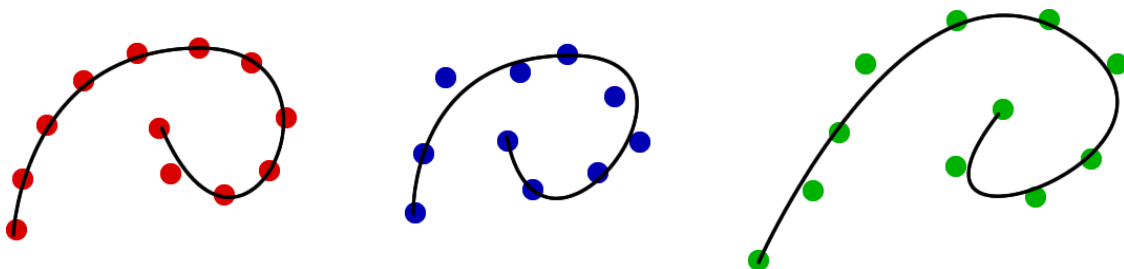


Figura 3.12. Repetição de gestos.

Alongando todas as marcas, tanto no eixo horizontal como vertical, é possível aproximar mais as semelhanças. As aproximações serão sempre relativas à primeira gravação, numa primeira fase é estudada a dimensão do gesto numa dinâmica de duas dimensões. Como representado na Figura 3.13, algumas representações vão aumentar outras diminuir.

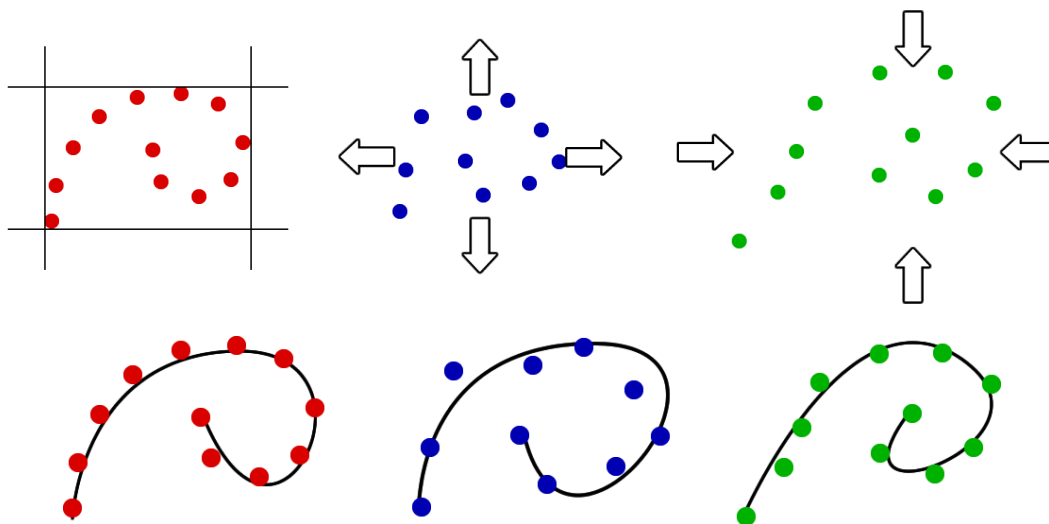


Figura 3.13. Redimensionamento da repetição de gestos.

Este redimensionamento aproxima mais os pontos das diferentes gravações, mesmo não coincidindo exatamente as trajetórias.

Com todos os pontos agora próximos de uma trajetória pretendida, pode-se aplicar um novo algoritmo que transforme todas as gravações num só gesto. Através das várias gravações é possível achar uma trajetória média e mais suave, este algoritmo é baseado essencialmente no cálculo da média móvel, representado com a fórmula (3.3).

$$MMS = \frac{p_M + p_{M-1} + \dots + p_{M-(n-1)}}{n} \quad (3.3)$$

Este cálculo é muito procurado na área de economia, para de análises de dados, é um dos métodos usados por investidores e operadores comerciais para auxílio da análise de tendência de preços. Esta média móvel dispõe de várias variantes com o intuito de se enquadrar em diversas áreas, neste caso, a variante utilizada será média móvel simples (MMS) (Ding, Liu, Pu, & Ni, 2013).

A Figura 3.14 (a) tenta representar três gravações do mesmo movimento, com a aplicação do algoritmo sugerido, baseado no cálculo da MMS, a solução será uma média ponderada com uma única trajetória de movimentos mais suaves, como na Figura 3.14 (b).

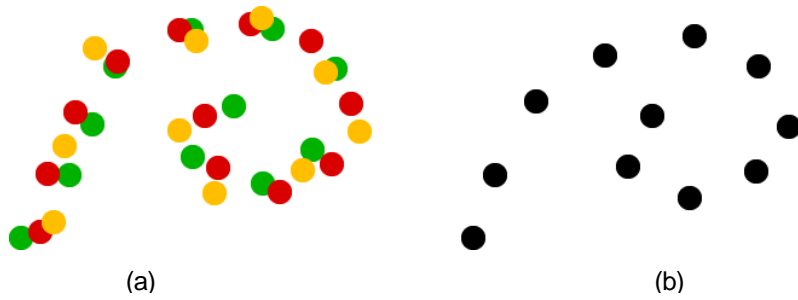


Figura 3.14. Aplicação da média móvel simples.

Transformação em equações

Por fim a fase que transforma todos os dados capturados numa forma de dados simplificada para ser guardada na biblioteca de gestos. Esta fase tem como objetivo transformar a sequência de pontos dos movimentos capturados num número reduzido de equações que o compoñham.

O grande desafio é como transformar estes pontos em equações polinomiais de grau 3, para tal é utilizada uma aproximação polinomial de grau 3 pelo método dos mínimos quadrados. O sistema de equações (3.4) mostra o método utilizado para o cálculo das equações polinomiais desta proposta. Obtido um dado conjunto de pontos de coordenadas (x, y) , são calculados os coeficientes a_0, a_1, a_2, a_3 através do cálculo inverso.

$$\begin{cases} a_0 \sum_{i=0}^n x_i^0 + a_1 \sum_{i=0}^n x_i^1 + a_2 \sum_{i=0}^n x_i^2 + a_3 \sum_{i=0}^n x_i^3 = \sum_{i=0}^n y_i x_i^0 \\ a_0 \sum_{i=0}^n x_i^1 + a_1 \sum_{i=0}^n x_i^2 + a_2 \sum_{i=0}^n x_i^3 + a_3 \sum_{i=0}^n x_i^4 = \sum_{i=0}^n y_i x_i^1 \\ a_0 \sum_{i=0}^n x_i^2 + a_1 \sum_{i=0}^n x_i^3 + a_2 \sum_{i=0}^n x_i^4 + a_3 \sum_{i=0}^n x_i^5 = \sum_{i=0}^n y_i x_i^2 \\ a_0 \sum_{i=0}^n x_i^3 + a_1 \sum_{i=0}^n x_i^4 + a_2 \sum_{i=0}^n x_i^5 + a_3 \sum_{i=0}^n x_i^6 = \sum_{i=0}^n y_i x_i^3 \end{cases} \quad (3.4)$$

Após o cálculo dos coeficientes é sempre feito um teste à qualidade da equação obtida, dentro do intervalo adequado, é testado se os pontos propostos para a equação se situam abaixo de uma distância d , pré-definida como a distância de erro máxima.

Como já referido anteriormente nesta proposta, o movimento nem sempre pode ser definido com apenas uma equação, existe o problema da inversão do sentido do movimento e também alguns movimentos com maior complexidade. Para isso é necessário algum pré processamento antes de tentar transformar todos os pontos numa equação.

Da lista de pontos obtida após a modelação do movimento é necessário verificar se em algum momento o sentido é invertido, caso isso se verifique, a lista de pontos é dividida. Considera-se o sentido invertido quando mais do que dois pontos seguem no sentido contrário. O número de divisões é o número de inversões que acontecem na lista de pontos.

É então com estas listas de pontos divididas que vamos tentar arranjar uma equação que defina o movimento descrito pelos pontos. Segue-se o problema dos movimentos complexos, no insucesso da

obtenção de uma equação o conjunto de pontos é dividido em duas partes iguais de forma a tentar simplificar as equações. Como tenta mostrar a Figura 3.15 (a), a equação a preto não consegue descrever os pontos a vermelho, segundo o raciocínio descrito anteriormente, o conjunto de doze pontos será dividido em dois conjuntos de seis pontos, os seis primeiros e os seis últimos. Na Figura 3.15 (b), a equação a azul já descreve o primeiro conjunto de seis pontos, mas novamente sem sucesso no último conjunto. A primeira equação (Azul) será guardada e os últimos seis pontos seguem, sucessivamente o mesmo raciocínio até que as equações calculadas tenham sucesso da descrição do movimento.

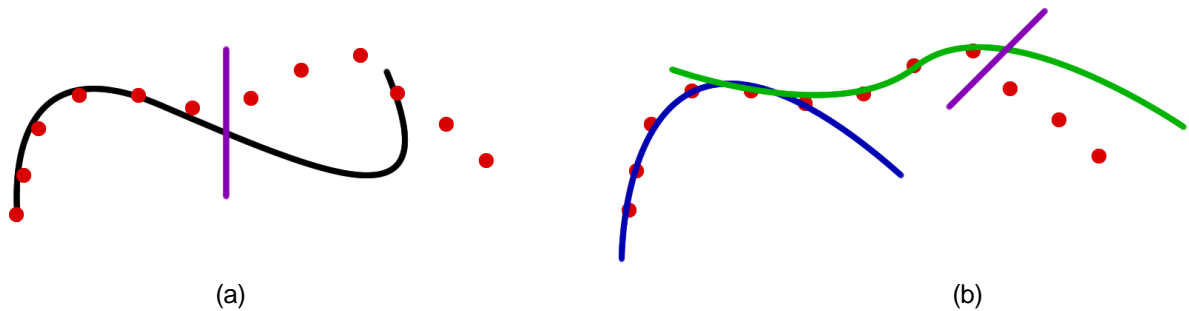


Figura 3.15. Divisão de equações.

Após a obtenção das equações necessárias para a definição do gesto, a classe de gesto encontra-se preparada para ser adicionada à biblioteca de gestos. Será então uma grande quantidade destes gestos que definirá a biblioteca de gestos.

3.3 Ferramentas de Suporte

Criar um gesto sem qualquer tipo de *feedback* torna-se um tanto mais complicado, por isso foram desenvolvidas algumas ferramentas de suporte à criação de gestos e algumas também para o reconhecimento dos gestos.

Uma das ferramentas disponíveis é uma janela de pré-visualização que mostra o que está a ser captado pela camara RGB do Kinect. Além da imagem RGB visualizada nesta janela, é possível também adicionar uma lista de articulações que pretendemos observar. A posição de cada articulação é seguida e apresentada através de um círculo vermelho, como mostra a Figura 3.16, neste caso foi adicionada uma lista com as articulações das mãos.

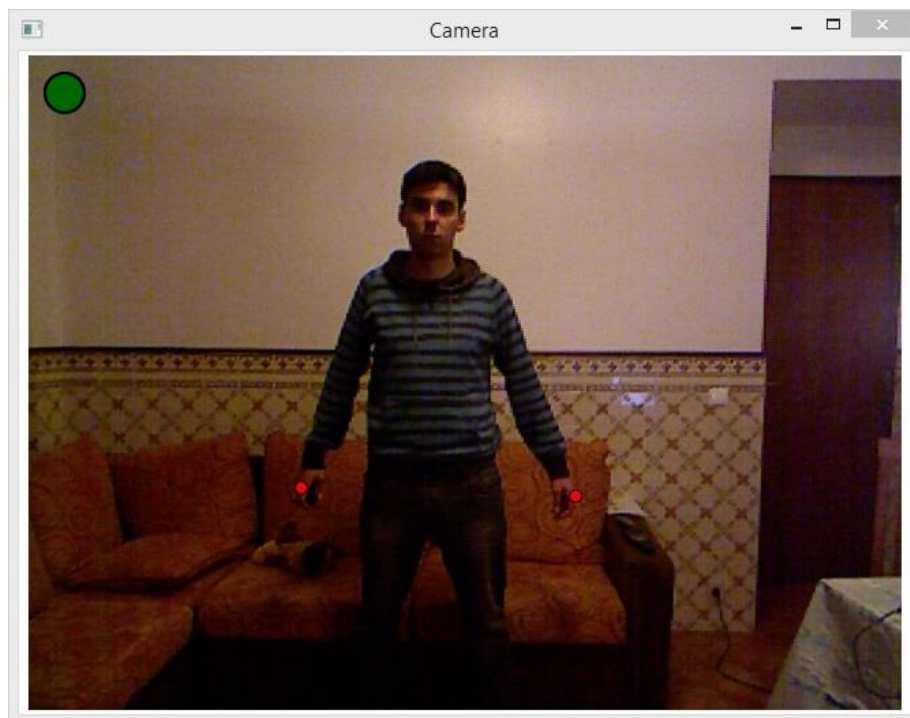


Figura 3.16. Imagem capturada da camara de pré-visualização.

Na camara de pré-visualização existe um indicador de gravação, este indicador representa-se por um círculo no canto superior esquerdo. Este círculo muda de cor consoante o estado, quando é iniciada uma gravação passa a vermelho, caso contrário estará sempre a verde como na Figura 3.16.

Uma outra ferramenta, similar, é uma janela que mostra apenas as posições das articulações. Esta janela, ilustrada na Figura 3.17, usa um outro tipo de captação do esqueleto, reconhecimento através da camara de profundidade.

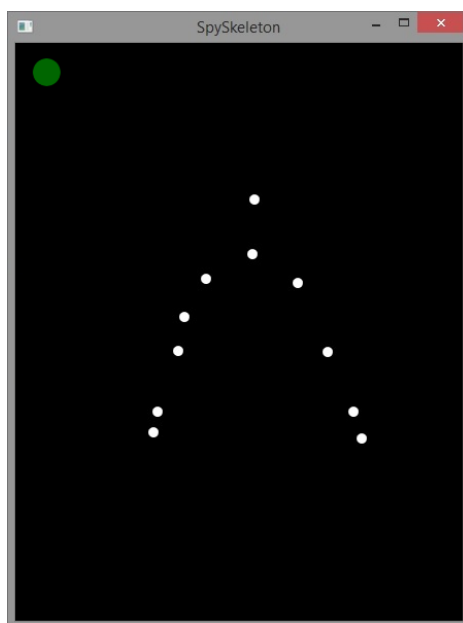


Figura 3.17. Pré-visualização das articulações através da camara de profundidade.

Esta janela segue o mesmo procedimento na inclusão de uma lista de articulações, mas as articulações aqui representadas, através de pontos brancos, não representam a posição absoluta mas sim uma posição relativa. Todos os pontos representam a distância a um ponto central designado por *ShoulderCenter*. Existe também uma indicação de quando o gesto se encontra em gravação.

A próxima ferramenta trata-se de uma janela de edição, esta edição ocorre depois dos pontos serem captados durante um intervalo de tempo e pode ser feita em cada uma das articulações pertencentes ao gesto. A Figura 3.18 é um exemplo dos pontos captados em relação à mão direita, como o título da janela indica. A grande utilidade desta ferramenta é poder-se redefinir onde queremos que o movimento comece e onde acabe, no canto superior encontra-se uma barra de deslizamento que permite essa redefinição. Para finalizar a edição o utilizador deve carregar no botão de salvar.

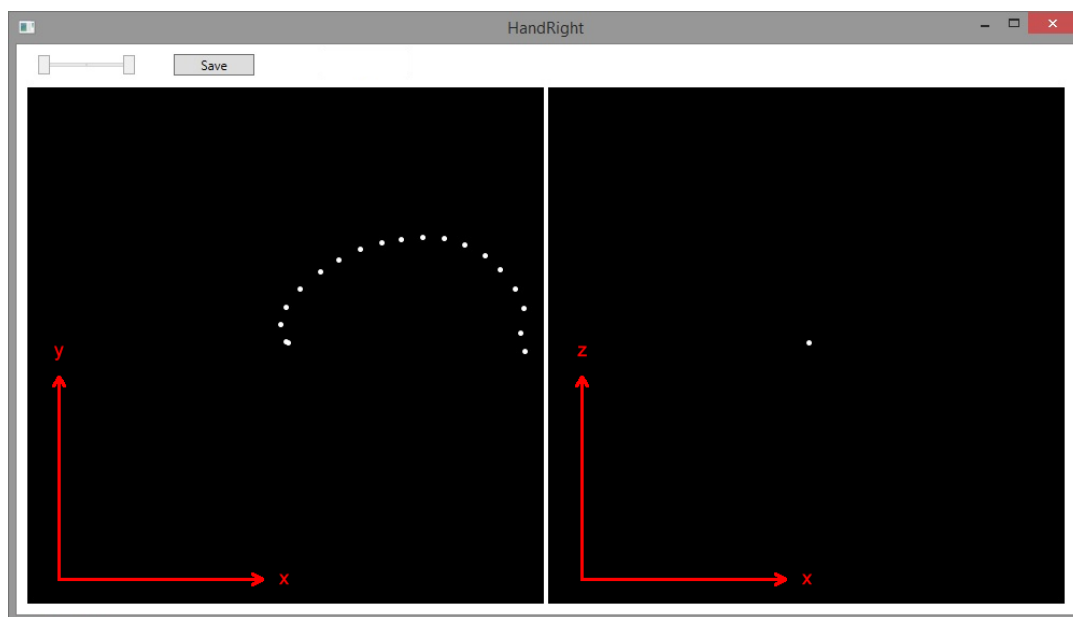


Figura 3.18. Janela de edição dos pontos capturados pelo Kinect.

Após esta redefinição de pontos o utilizador pode fazer mais gravações e novamente editar os pontos, posteriormente ocorre a definição das equações com estes pontos editados ou não. E surge novamente uma ferramenta que permite a visualização das equações obtidas, caso o utilizador pretenda ver o resultado. A janela de visualização de equações (Figura 3.19) permite ao utilizador uma visualização a três dimensões. Através do controlo do rato é possível rodar a equação de forma a verificar se era este o movimento pretendido.

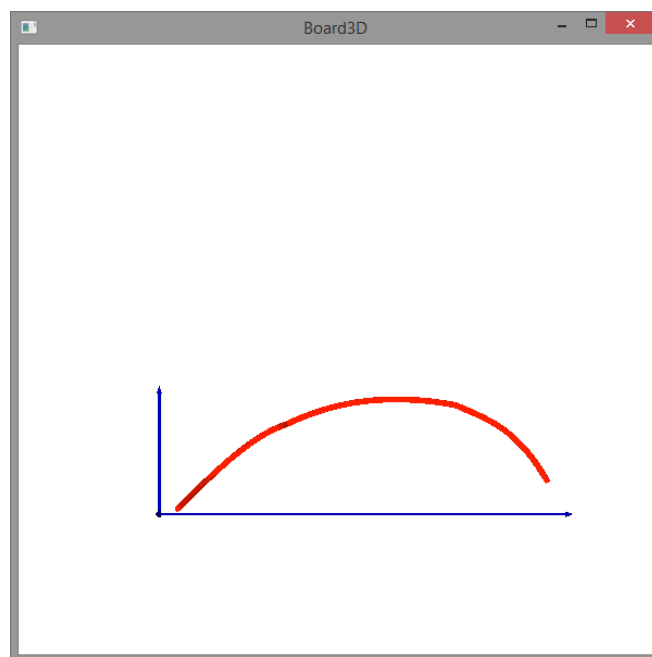


Figura 3.19. Janela de visualização da equação em três dimensões.

Até à visualização desta janela de equações é possível descartar o gesto, se o utilizador assim desejar poderá inserir a equação, representante do gesto, na biblioteca.

Após a composição da biblioteca desejada e o reconhecimento ativado, além de serem disparados eventos com os gestos é possível ver quais os gestos reconhecidos. Esta janela de apoio ao reconhecimento é composta por duas partes, como mostra na Figura 3.20. A parte da direita é um histórico com todos os gestos que foram reconhecidos, sendo o primeiro o mais recente. Na outra metade da janela encontram-se os gestos disponíveis na biblioteca e surge também cada articulação associada ao gesto. Tanto parte do gesto (uma articulação) como todo o gesto se for reconhecido é alterada a cor. A Figura 3.20 foi capturada quando se verificou o reconhecimento de um gesto, designado por Direita, que corresponde a um movimento para a direita com a mão direita, alterando a cor, no texto, do nome do gesto. Paralelamente, esse movimento com a mão direita corresponde também a parte de outro gesto (Esticar), assinalado a vermelho o reconhecimento dessa parte do gesto.

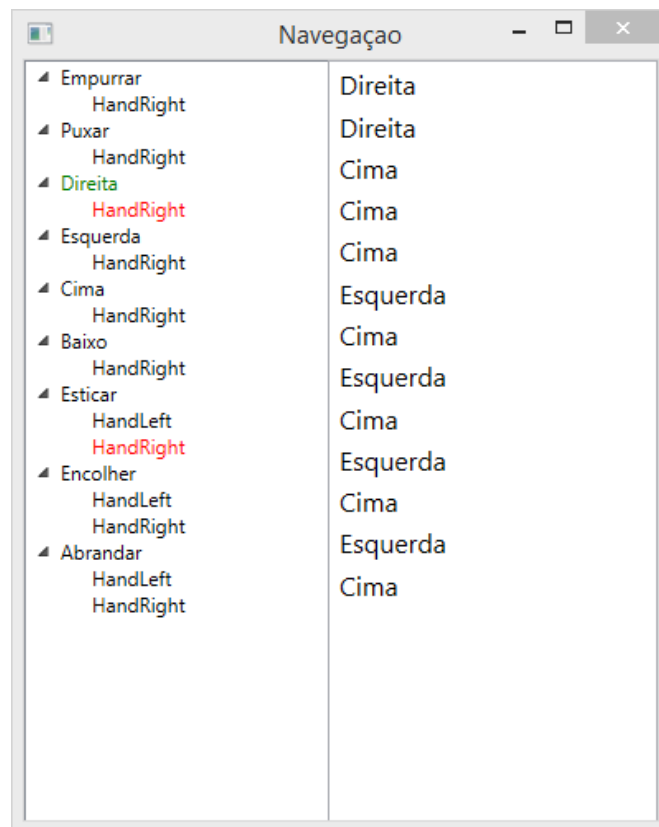


Figura 3.20. Lista de gesto da biblioteca e histórico de reconhecimento.

3.4 Reconhecimento em Tempo Real

Esta biblioteca de gestos pode-se distinguir em dois modos, um modo passivo que engloba tudo o que já foi falado anteriormente sobre a criação dos gestos e algumas ferramentas de suporte e um modo ativo. Este modo ativo corresponde ao reconhecimento em tempo real dos gestos presentes na biblioteca. Pode ser ativado e desativado sempre que o utilizador desejar.

Em primeiro lugar é necessário ter uma estrutura que possa armazenar todos os pontos ao longo do tempo, essa estrutura como já mencionado neste capítulo vai ser uma lista de elementos indeterminados. Assim que ativado o reconhecimento começarão a ser adicionados os vários pontos do esqueleto, como forma de histórico de pontos, e posteriormente removidos assim que o sistema reconhecer que não pertencem a nenhum gesto.

Um dos algoritmos mais básicos deste reconhecimento é a comparação dessa lista de pontos com um dos gestos armazenados na biblioteca, a comparação é feita com todos os pontos presentes no histórico de pontos da respetiva articulação, como na imagem da Figura 3.21, os pontos são testados se ficam próximos da equação (linha a preto) com uma distância máxima de erro, caso seja maior que essa distância, como no último ponto do exemplo (ponto a vermelho), verifica-se que os pontos não correspondem a este gesto. Outra das verificações é a tendência dos pontos, se os pontos não seguirem o sentido da equação o gesto é esquecido.

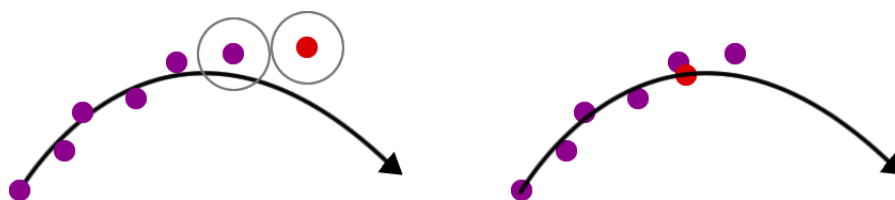


Figura 3.21. Erro no reconhecimento.

Este caso anterior mostra como é comparado apenas uma articulação de um gesto, no caso de o gesto ser composto por mais do que uma articulação, vários testes semelhantes são efetuados para essas articulações com a respetiva equação. Se alguma das articulações invalidar o gesto, o resto das articulações continuam a ser comparadas, funcionando assim de forma independente.

Durante a comparação dos pontos, do histórico, com a distância à equação se surgir um ponto próximo da extremidade da equação, parte do gesto é dada como válido e é anotado o momento em que esta validação surgiu. Para o caso de o movimento de uma dada articulação ser definido por mais que uma equação, essas validações passam por todas as equações de forma ordenada e apenas é validado quando chegar ao final da última equação.

Para o gesto ser completamente validado e no caso ser composto por mais de uma articulação, ele terá um tempo limite para que todas as articulações sejam dadas como válidas. Após uma das articulações der dada como válida é feita uma verificação em todas as outras articulações do gesto e é verificado se todas essas articulações foram reconhecidas durante esse intervalo de tempo definido. Este espaço temporal foi definido em três segundos após a realização de alguns testes com os gestos. A Figura 3.22 mostra como funciona a dimensão do tempo em cada articulação, cada cor representa uma articulação pertencente ao gesto e o seu comprimento o intervalo de tempo a que está habilitado o reconhecimento. No exemplo da esquerda existe um momento em que todas as articulações estão habilitadas ao reconhecimento, esse momento é identificado com a linha vertical a preto, já no exemplo à direita o tempo de reconhecimento expira assim que acaba o período da primeira articulação a ser reconhecida, neste caso o traço a vermelho. No primeiro caso há sucesso, no segundo há falha no reconhecimento.



Figura 3.22. Dimensão de tempo no reconhecimento de um gesto.

Por vezes os pontos presentes no histórico não invalidam um dado gesto mas também não são suficientes para atingir a extremidade da equação, neste caso nada acontece e o sistema continua a

acumular pontos do esqueleto. Sempre que é captado um ponto pelo Kinect esse ponto é adicionado ao histórico de pontos da respectiva articulação para mais tarde ser validado, esta validação ocorre consoante a frequência definida no temporizador da biblioteca de gestos e acaba com um dos três fins em cada articulação:

- Detecção de uma parte do gesto.
- Invalidação da sequência de pontos na respectiva articulação.
- Insuficiência de pontos.

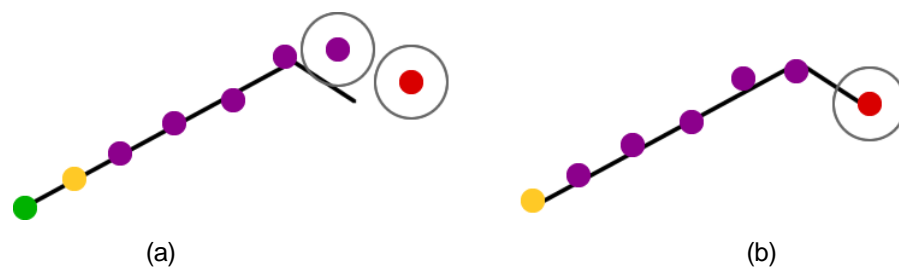


Figura 3.23. Remover pontos.

Caso a sequência de pontos testada invalide o movimento da articulação, a articulação volta a ser testada assim que o temporizador da biblioteca de gesto volte a disparar, mas antes de ser descartado o movimento o ponto mais antigo será removido do respetivo histórico de pontos.

No exemplo da Figura 3.23 (a) o ponto a vermelho já não corresponde à margem permitida pela função, como explicado anteriormente, o ponto mais velho (ponto verde) é removido e o próximo reconhecimento começará no segundo ponto (ponto amarelo). No exemplo escolhido, o ponto que anteriormente invalidou o reconhecimento no próximo reconhecimento será coincidente com o fim da equação (Figura 3.23 (b)), validando assim parte do gesto.

No núcleo da biblioteca de gestos são executados alguns processos essenciais para o sincronismo deste reconhecimento. Após a ativação do reconhecimento as coordenadas de cada articulação são reenviadas para cada articulação que pertence ao gesto, mas apenas aquelas que estão habilitadas ao reconhecimento, as outras serão esquecidas, como ilustrado na Figura 3.24. Este processo é repetido para cada gesto existente na biblioteca. Paralelamente e a uma frequência contante é ativado o reconhecimento de cada articulação para os pontos adquiridos até ao momento. Esta frequência é definida acima da frequência a que o Kinect adquire os pontos, não fazendo sentido reconhecer novamente o movimento sem novos pontos.

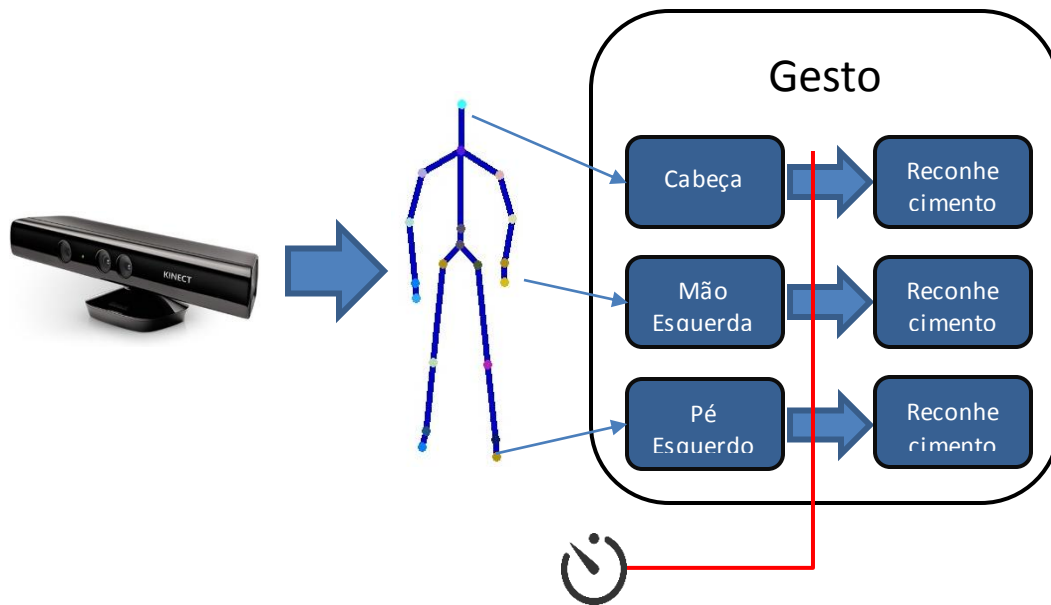


Figura 3.24. Representação do processo executado quando ativado o reconhecimento.

Estes processos repetem-se múltiplas vezes, cada um à sua frequência, até que seja parado o reconhecimento.

3.5 Conclusão

Neste capítulo foram descritos os passos necessários para efetuar a gravação e reconhecimento de gestos. Foram também apresentadas algumas ferramentas que permitem ter uma maior noção, quer na gravação, quer no reconhecimento.

A Figura 3.25 resume todo o processo proposto para a gravação de um gesto. Inicialmente é feita uma limpeza dos pontos captados, de forma a obter informação mais precisa e a redução de alguma informação desnecessária. Após as gravações desejadas, o conjunto de informação é normalizado para haver uma maior semelhança entre todas as gravações. Com esta semelhança aplica-se então uma suavização, através da MMS, para se obter apenas um conjunto de pontos. Com este último passo fica concluído o tratamento de dados para uma modelação através de equações, tarefa que conclui o processo.

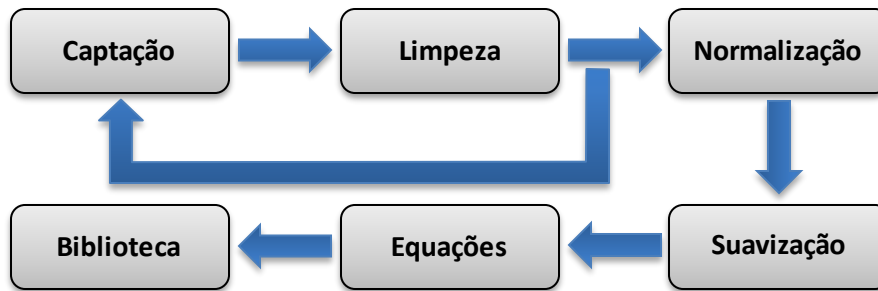


Figura 3.25. Diagrama do processo de gravação do gesto.

Com este processo de gravação concluído o processo de reconhecimento fica habilitado.

Um dos principais desafios foi a forma de como trabalhar em tempo real. Para tal, a solução proposta é tratar cada gesto de forma independente. À medida que os dados de cada articulação são acumulados, essa informação é “enviada” para cada um dos gestos na biblioteca e aí é verificada a correspondência. Este método permite a identificação do gesto, de imediato, assim que o último momento é captado e se nenhum dos anteriores o invalidar.

Além dos processos mencionados, a proposta inclui algumas ferramentas úteis. Estas ferramentas foram pensadas para ajudar o utilizador na visualização e definição de gestos, uma das quais fundamental para definir os pontos desejados (limites) para a equação.

4. Validação

Por fim são apresentados alguns resultados de forma a mostrar a eficácia do modelo proposto. São mencionadas as condições de funcionamento e os gestos executados.

Numa primeira fase é mostrado como alguns algoritmos podem melhorar a definição do gesto. A segunda parte do teste mostra a precisão do reconhecimento.

4.1 Material e Métodos

Os mecanismos e algoritmos de gravação e reconhecimento de gestos foram desenvolvidos para o SDK do Kinect, como já mencionado anteriormente. De referir também que o Kinect utilizado foi a versão para a consola Xbox, ligeiramente diferente da versão para Windows que apresenta algumas melhorias no seu *hardware* (Jana, 2012). Outros dos aspetos relevantes para o desempenho dos algoritmos é velocidade de processamento, neste caso foi usado um processador i7 com uma frequência de 1.9 GHz.

Para um correto funcionamento do Kinect foi colocado a sensivelmente 2 metros de distância do utilizador e a uma altura de 1.4 metros como mostra na Figura 4.1, de modo a tentar captar todo o esqueleto.

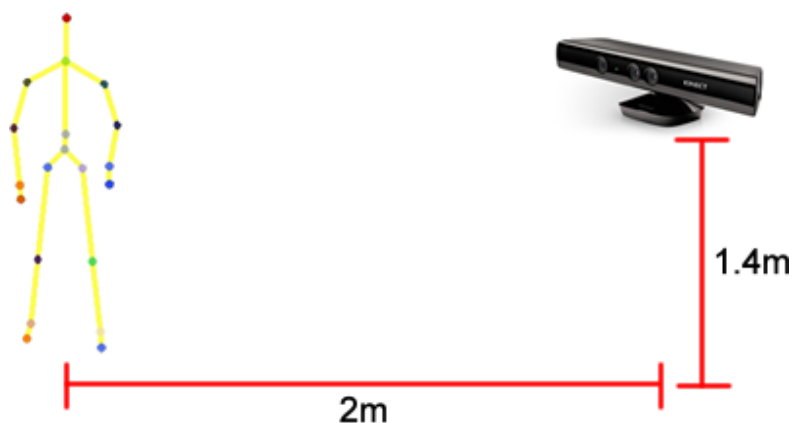


Figura 4.1. Disposição do cenário usado para teste.

Foram executados vários gestos, estudados e explicados no próximo subcapítulo, de forma a mostrar a fiabilidade do sistema dando a conhecer quais os limites destes algoritmos. Para estes testes foi utilizado um grupo de 8 pessoas, variando o número de amostragens consoante o teste realizado.

Os principais pontos a testar nesta biblioteca de gestos são os seguintes:

- Composição das funções na gravação de gestos.
- Fiabilidade no reconhecimento singular de gestos simples de apenas uma articulação.
- Fiabilidade no reconhecimento singular de gestos com mais de uma articulação.
- Fiabilidade no reconhecimento singular de um grupo de gestos com alguma complexidade.
- Reconhecimento de vários gestos integrados numa biblioteca de gestos.

Na primeira parte, gravação de gestos, pretende-se mostrar como surgem as equações, como elas ficam visualmente e as melhorias que vão sofrendo com os algoritmos aplicados. Como indicado na Figura 4.2, os módulos observados na validação são a Normalização de todos os conjuntos de pontos captados em relação ao gesto, a aplicação de um filtro a este conjunto de pontos referidos e as equações finais, estes módulos são aplicados depois de uma edição do gesto captado e anteriormente a serem guardados na biblioteca de gestos.

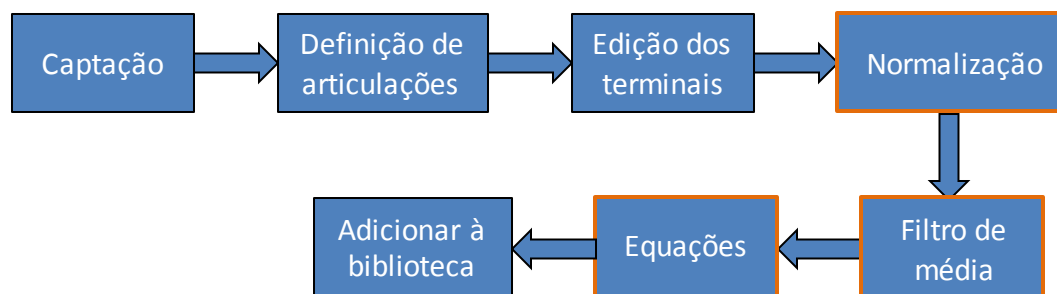


Figura 4.2. Indicação dos módulos observados na validação.

No que diz respeito ao reconhecimento de gestos, numa primeira fase os gestos vão ser avaliados um a um, com isto pretende-se testar a fiabilidade de cada gesto quando não existe mais a interferir. Nesta avaliação singular são ainda testadas as capacidades de reconhecimento de quatro grupos de gestos distintos, apresentados no subcapítulo seguinte.

Como validação final, são testados vários gestos inseridos numa só biblioteca. Aqui é possível testar o reconhecimento dos vários gestos com uma espécie de ruído à mistura, por outras palavras, com vários gestos habilitados ao reconhecimento. Pretende-se com este último processo verificar se ao executar o gesto X ele é detetado ou se algum outro gesto é reconhecido.

4.2 Gestos

Os gestos escolhidos foram pensados de modo a tentar avaliar o sistema da melhor forma possível. Estes gestos foram divididos em quatro categorias diferentes, a primeira categoria (Grupo I) é composta por gestos apenas com uma articulação (Figura 4.3). Estes gestos são normalmente usados no controlo de aplicações, descrevem a forma mais simples do que pode ser um gesto e são aproximações de vetores nos dois sentidos de cada um dos eixos tridimensionais. A representação dos gestos mostrado

nas próximas figuras seguem o seguinte código de cores, vermelho para a mão direita e verde para a mão esquerda.

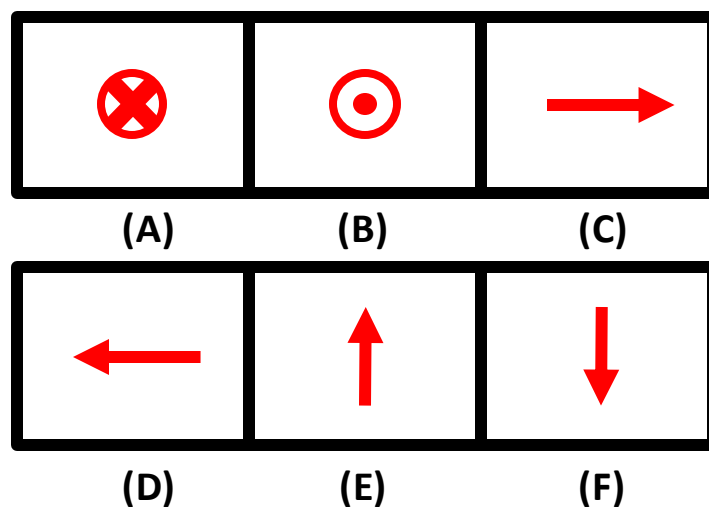


Figura 4.3. Gestos simples utilizados no teste da aplicação. (Grupo I)

O segundo grupo de gestos é composto novamente por gestos simples (Figura 4.4), de apenas uma direção, mas desta vez por mais do que uma articulação. Este grupo (Grupo II) de gestos pretende testar a deteção de movimentos em simultâneo, isto é, gestos compostos por duas ou mais articulações.

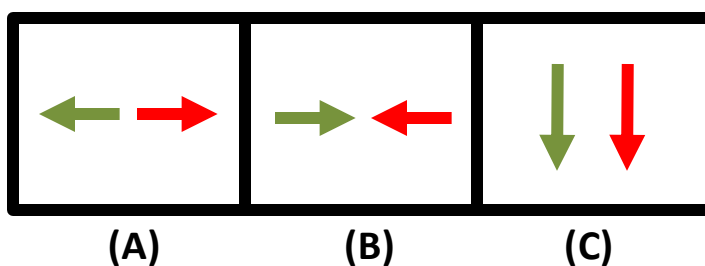


Figura 4.4. Gestos simples de várias articulações utilizados no teste da aplicação. (Grupo II)

Por fim, o último conjunto (Grupo III) é usado para testar a fiabilidade do sistema com gestos de maior complexidade. Aqui é também testada a capacidade da criação de equações para gestos complexos, estes são pensados de maneira a tentar complicar, matematicamente, as equações. Forçando também a utilização de múltiplas equações na mesma articulação.

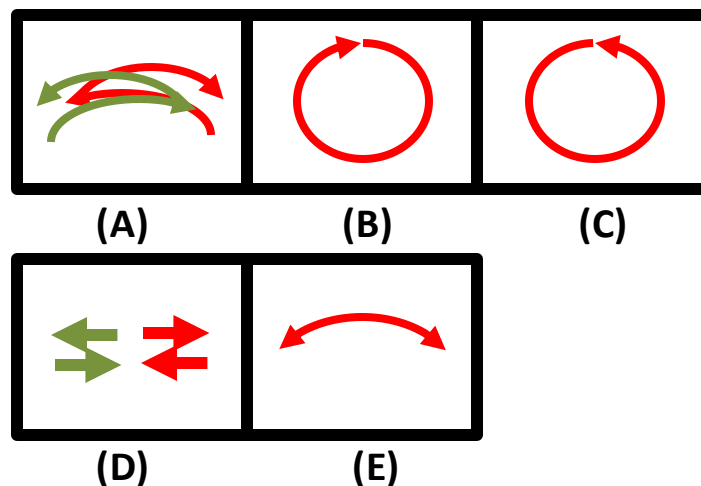


Figura 4.5. Gestos complexos de várias articulações utilizados no teste da aplicação. (Grupo III)

Nestes gestos (Figura 4.5) foram tidos em conta vários movimentos que se usam no dia-a-dia, por exemplo o gesto A é um movimento que as pessoas costumam utilizar para mandar parar alguém ou sinalizar que algo acabou, o gesto D representa as palmas, apesar de só estar representado um movimento para cada lado é necessário fazer este movimento 3 vezes para ser reconhecido, de forma a aumentar a complexidade.

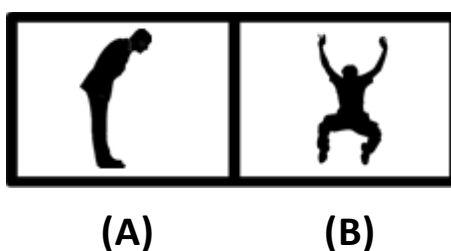


Figura 4.6. Grupo de gestos com outros pontos do corpo. (Grupo IV)

Este último grupo de gestos (Figura 4.6) é um complemento aos anteriores, no caso da Figura 4.6 (A) o gesto é a execução de uma vénia, este gesto é definido por três pontos (cabeça, ombro direito e ombro esquerdo), enquanto o gesto da Figura 4.6 (B) é um salto sendo por quatro pontos (ombro direito, ombro esquerdo, pé direito e pé esquerdo).

Estes dois gestos permitem o teste a outras articulações diferentes das duas mãos, que são usadas com uma maior frequência.

4.3 Resultados

4.3.1 Gravação do Gesto

Como representado na Figura 4.2 a criação de um gesto começa com a captação de todas as articulações durante um certo intervalo de tempo, depois disso são definidas as articulações que integram o gesto e em cada uma é indicado onde começa e acaba. Após esta edição, caso exista mais que uma captação do mesmo gesto, é feita uma normalização explicada no capítulo anterior. Este conjunto de captações normalizadas associadas ao filtro de média provocam melhorias significantes para posterior definição das funções. Para uma melhor percepção, os gestos foram executados sem variar a dimensão de profundidade. De seguida são apresentadas algumas imagens captadas durante a criação do gesto.

Foram feitas 4 gravações do mesmo gesto, de seguida foram obtidas as equações para o conjunto de pontos da primeira gravação e seguidamente as equações para o conjunto de pontos da média de todas as gravações.

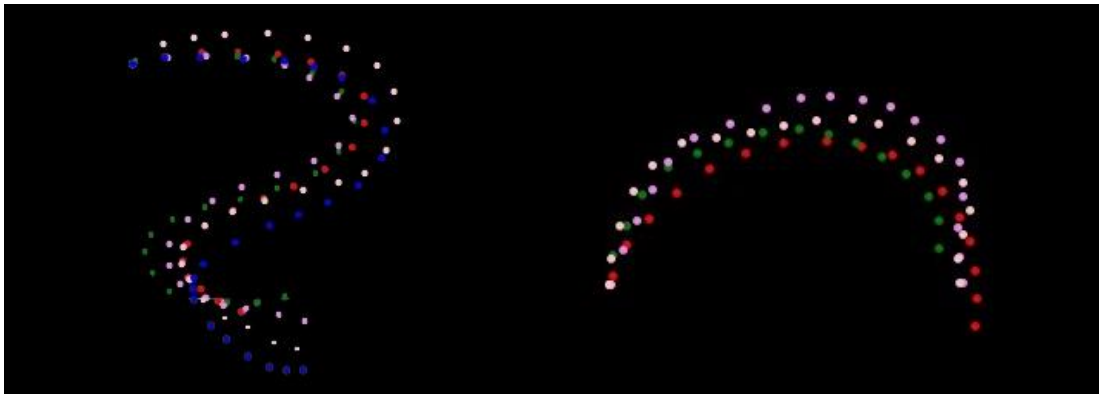


Figura 4.7. Múltipla captação de pontos de 2 gestos diferentes.

A Figura 4.7 corresponde à captação das 4 gravações de 2 gestos distintos, visivelmente perceptíveis, estas imagens são captadas posteriormente à normalização. De seguida são obtidas as funções que descrevem a trajetória de um dos conjuntos de pontos (o primeiro), o resultado dessas equações é ilustrado na Figura 4.8.

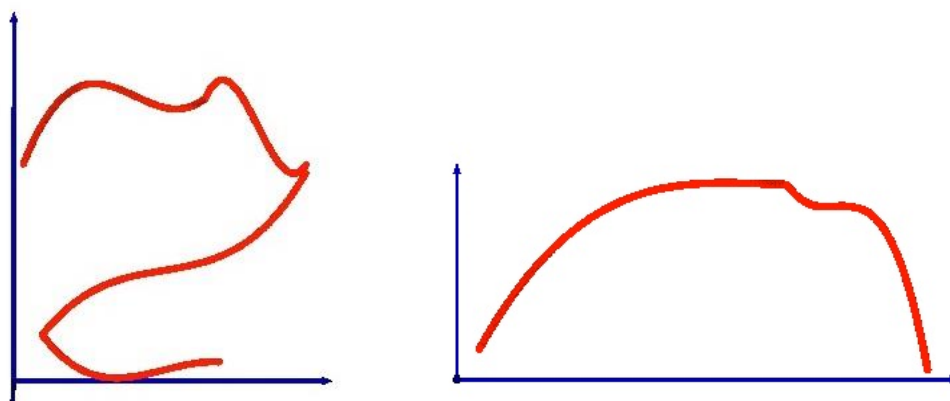


Figura 4.8. Definição do gesto por equações utilizando a primeira captação de dados.

Estas representações dos gestos definidas pelo conjunto de equações assemelham-se aos pontos captados e representados na Figura 4.7, mostrando que é possível proceder ao reconhecimento de dados apenas com uma captação. O grande problema de usar apenas uma captação é a existência de pontos indesejáveis e que não correspondem ao movimento feito pelo utilizador, por outras palavras, um reconhecimento deficiente do esqueleto por parte do Kinect. De notar ainda que nesta representação (Figura 4.8) um dos gestos utiliza 4 equação para a definição do gesto, número este que pretende ser reduzido com a aplicação do filtro de média aplicado no conjunto de captações.

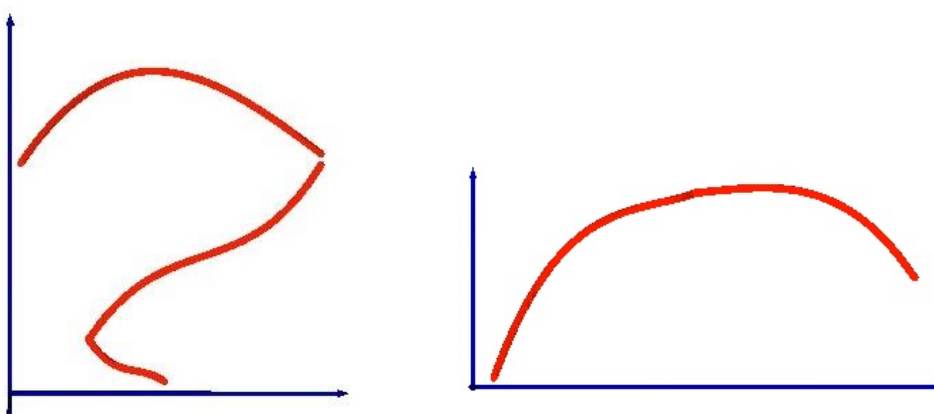


Figura 4.9. Definição do gesto por equações utilizando um filtro no conjunto das captações.

Através da aplicação do filtro de média, nota-se melhorias da definição do gesto na Figura 4.9. Um dos gestos passa a ser definido por menos equações e ambos têm agora uma trajetória mais suave permitindo uma melhor definição da margem de erro na parte do reconhecimento do gesto.

4.3.2 Reconhecimento

No primeiro teste, um grupo de 4 pessoas executou os gestos do Grupo I, estes são gestos simples que corresponde a uma só articulação, a mão direita. Foi pedido que os gestos fossem executados a uma velocidade moderada devido ao poder de processamento exigido pelo Kinect.

Cada pessoa representou cada gesto 10 vezes, sendo anotado o número de vezes que o sistema detetou os gestos. Os resultados são apresentados na Tabela 4.1.

Tabela 4.1. Número de gestos reconhecidos do Grupo I.

	Pessoa I	Pessoa II	Pessoa III	Pessoa IV	Total
Gesto A	4	2	5	4	15/40
Gesto B	4	4	4	3	15/40
Gesto C	9	8	10	10	37/40
Gesto D	10	10	10	10	40/40
Gesto E	10	9	9	10	38/40
Gesto F	9	6	7	10	32/40

No teste aos gestos do Grupo I nota-se um melhor reconhecimento nos gestos em que a coordenada de profundidade é irrelevante. Os gestos C e D são já detetados pelo Kinect, numa comparação entre os dois, pode-se dizer que a taxa de deteção neste teste é muito idêntica à taxa de deteção do reconhecimento integrado no SDK oficial.

O segundo teste ao reconhecimento, desta vez a gestos do Grupo II, foi experimentado com o mesmo número de amostras de maneira a se poder fazer algumas comparações com os gestos do Grupo I. Este teste tem como objetivo mostrar a capacidade de reconhecimento com mais do que uma articulação, os gestos são basicamente os mesmos que o anterior, só que desta vez são necessários serem executados 2 em simultâneo. A Tabela 4.2 mostra os resultados obtidos.

Tabela 4.2. Número de gestos reconhecidos do Grupo II.

	Pessoa I	Pessoa II	Pessoa III	Pessoa IV	Total
Gesto A	8	10	7	10	35/40
Gesto B	9	6	2	8	25/40
Gesto C	10	10	10	10	40/40

A Tabela 4.2 mostra uma taxa de reconhecimento idêntica à Tabela 4.1, a execução de movimentos de duas articulações não altera a taxa de sucesso.

O principal sucesso do reconhecimento destes movimentos simples está na definição das equações, quanto mais simples e precisas, melhor é reconhecido. Esta definição pode-se complicar quando o movimento dos gestos é composto por muitas inversões de sentido e curvas apertadas.

O próximo grupo de gestos (Grupo III) é um conjunto de vários gestos ditos complexos. Pretende-se reconhecer alguns gestos cujas características sejam variações de sentido e com algumas curvas. Estes gestos foram testados num primeiro momento individualmente e com o mesmo número de amostras (40, 10 por cada pessoa). A Tabela 4.3 mostra as taxas de reconhecimento conseguidas.

Durante o processo de gravação destes gestos mais complexos (Grupo III), o método proposto revelou muita dificuldade na composição das equações. Esta dificuldade foi causada pela dimensão de profundidade, ou seja, a coordenada z . Por isso, nestes testes a validação da coordenada z não foi tida em conta. O problema deve-se ao subconjunto de pontos usado para a definição z em função de x ser o mesmo que o conjunto de pontos usado na definição da função de y em função de x , o que pode ser o conjunto ideal para uma equação pode não ser para a outra.

Tabela 4.3. Número de gestos reconhecidos do Grupo III.

	Pessoa I	Pessoa II	Pessoa III	Pessoa IV	Total
Gesto A	6	9	7	7	29/40
Gesto B	10	7	8	8	33/40
Gesto C	6	8	9	7	30/40
Gesto D	10	8	5	7	30/40
Gesto E	10	7	4	9	30/40

Ignorando o problema na criação dos gestos, na aquisição da equação com coordenadas de profundidade, a taxa de reconhecimento dos gestos a duas dimensões é satisfatória. Verificou-se que o reconhecimento tem alguns problemas quando algumas articulações se encontram ou se cruzam. Este problema é criado devido ao reconhecimento do esqueleto e à técnica usada para seguir as articulações.

Como teste final a este método de reconhecimento em tempo real, com gestos criados através da gravação, foi criada uma biblioteca de gestos com todos os gestos do grupo III. Com todos estes gestos habilitados a serem reconhecidos pretende-se testar a capacidade de decisão.

Cada pessoa executará alguns gestos e será observado o número de vezes que o gesto correto é reconhecido e o número de vezes que é reconhecido outro gesto na tentativa de representar o correto.

Quando nenhum gesto for reconhecido é considerado gesto nulo e todos os gestos nulos serão ignorados para esta estatística, sendo que essa característica já foi avaliada nos testes anteriores. O objetivo resume-se a reconhecer o gesto correto no meio de ruído. Este procedimento permite avaliar a capacidade de diferenciação de entre os vários gestos.

Apenas alguns gestos vão ser testados no reconhecimento, mas todos serão incluídos na biblioteca. Cada gesto foi tentado 50 vezes, sendo este valor dividido por mais de uma pessoa.

À medida que foram efetuados alguns testes de reconhecimento notou-se que a biblioteca de gestos tornava-se lenta ao reconhecer um número considerável de gestos em simultâneo. Não se chegou a número preciso para o bom funcionamento da biblioteca, mas em alguns casos, reconhecer 8 gestos em simultâneo causou atrasos no mapeamento das articulações, este problema relaciona-se também com eficácia do processador usado.

O procedimento de teste é semelhante ao anterior, cada pessoa vai executar novamente o mesmo número de gestos e serão verificados quantos são reconhecidos corretamente.

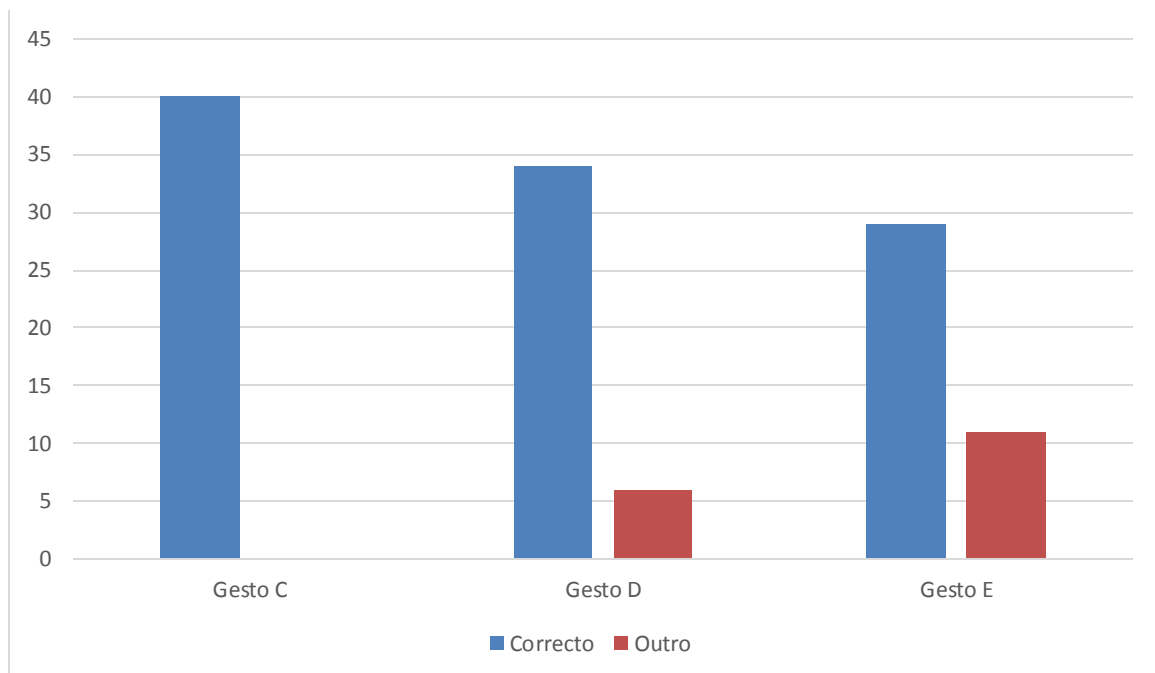


Figura 4.10. Taxa de sucesso de reconhecimento de gestos inseridos na biblioteca (Grupo III).

Na Figura 4.10 um dos gestos apresenta uma taxa de reconhecimento ótima (Gesto C), significa que este gesto nunca é confundido com nenhum dos outros gestos presentes na biblioteca, já nos restantes gestos testados existe algum erro. Durante o teste do gesto E houve um gesto que se evidenciou ao ser reconhecido de forma incorreta, o gesto C.

No decorrer desta realização experimental verificou-se que algumas articulações não funcionam bem neste método de criação e reconhecimento de gestos. Este defeito deve-se à captação de pontos

errados e movimentos de curta distância, as equações raramente correspondem ao movimento pretendido. Outro dos problemas, já referenciados, é a enorme capacidade de processamento necessária para o reconhecimento. Este problema é denunciado de forma muito clara quando existe um grande conjunto de gestos na biblioteca.

No que diz respeito à criação do gesto o maior problema surge na falta de eficácia da informação correta dada pelo Kinect, as coordenadas das articulações. Mesmo com aplicação do filtro de média, uma captação do gesto fora do normal pode “estragar” a composição das equações. Outro problema encontrado durante a realização da parte experimental foi a criação das equações que incluem a componente de profundidade, estas equações eram obtidas através do conjunto de pontos usado para composição da equação de y em função de x , sendo visível que não era o conjunto adequado quando foram criados os gestos do grupo III.

Definição manual das equações.

Para um real teste ao algoritmo de reconhecimento e de forma independente, da gravação de gestos, foram criadas algumas equações de forma manual de forma a ultrapassar alguns erros mencionados. Alguns dos gestos utilizados são idênticos aos já anteriormente testados, como os gestos C, D, E e F do Grupo I, o gesto A do Grupo II e os gestos B, D e E do grupo III, de forma a complementar melhor o teste foram ainda definidos mais dois gestos (Grupo IV). Os gestos mencionados foram definidos apenas por um conjunto de equações quadradas e algumas retas.

Este teste permite também mostrar resultados que podem ser atingidos caso a gravação de gestos seja melhorada.

O procedimento de testes é semelhante ao anterior já mencionado, sendo o primeiro teste referente à eficácia individual a cada um deles.

Tabela 4.4. Taxa de sucesso para as equações criadas de forma manual.

	Pessoa I	Pessoa II	Pessoa III	Pessoa IV	Total
G. I Gesto C	10	10	9	10	39/40
G. I Gesto D	10	10	10	10	40/40
G. I Gesto E	10	10	10	10	40/40
G. I Gesto F	9	10	10	10	39/40
G. II Gesto A	10	10	10	10	40/40
G. III Gesto B	10	7	6	8	31/40
G. III Gesto D	9	7	10	10	36/40
G. III Gesto E	10	9	8	8	35/40
G. IV Gesto A	9	10	8	9	36/40
G. IV Gesto B	10	10	9	8	37/40

Este ensaio, referente à Tabela 4.4, revelou-se com melhores taxas de sucesso devido à melhor simplicidade e melhor definição dos gestos. No entanto, a inclusão, no projeto, deste tipo de criação de equações (manual) pode não ser uma significativa mais-valia. Foi possível definir estes gestos porque não se trata de movimentos de grande complexidade. Para esta criação manual é necessário também ter uma grande experiência na noção de espaço, visto que a relação das coordenadas obtidas com as unidades da equação não é totalmente linear.

O ensaio permitiu apenas provar que o melhoramento da criação das equações pode permitir um melhor reconhecimento, por isso, a evolução poderá passar pelo melhoramento da criação automática das equações.

Em relação ao reconhecimento com vários gestos em simultâneo, foram escolhidos alguns dos anteriores para o ensaio. Esse ensaio é mostrado em forma de gráfico na Figura 4.11.

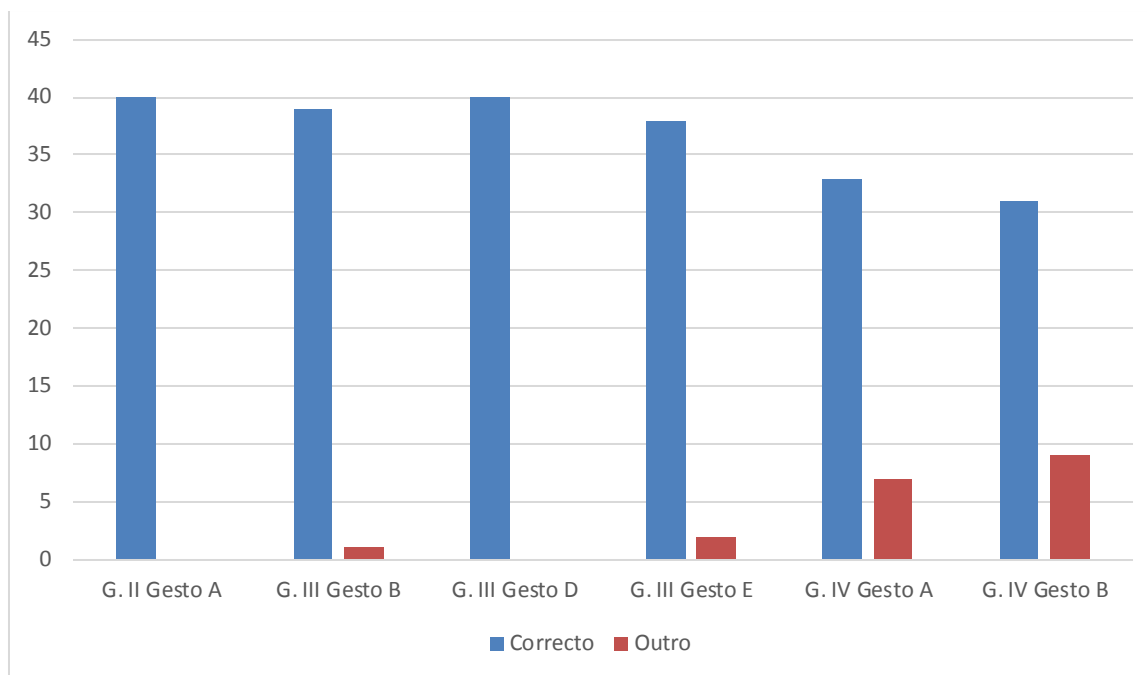


Figura 4.11. Taxa de reconhecimento, com equações criadas manualmente.

Durante a realização deste teste verificou-se que os gestos do Grupo IV eram confundidos um com o outro, este erro pode ocorrer devido à deficiência da posição dada pelo sistema, visto que em um dos gestos as articulações dos pés eram reconhecidas mesmo sem mexer os pés.

No caso do Gesto B do Grupo IV, a taxa de erro é aceitável visto que uma percentagem do Gesto A do mesmo grupo é executado, ao contrário do reconhecimento do Gesto A em relação ao B, que apenas deverá conseguir melhores taxas de reconhecimento com um melhoramento na captação da posição das articulações por parte do sistema, nos outros casos o reconhecimento mostra-se muito eficiente.

Em relação à definição do movimento com equações mais simples em vez das definidas inicialmente (equações de grau 3), ficou visível que é suficiente para descrever os gestos aqui ensaiados e teoricamente irá simplificar a criação das equações, o que seria muito interessante de verificar experimentalmente.

5. Conclusões e Trabalhos Futuros

Nesta dissertação desenvolveu-se um modelo de criação e reconhecimento de gestos para sensores de movimento.

De seguida é feita uma reflexão sobre o trabalho desenvolvido e os resultados obtidos, são também propostos novos caminhos a seguir.

5.1 Conclusão

As interfaces naturais apresentam-se como uma das principais formas de comunicação para os próximos tempos. Neste sentido começam a aparecer novos projetos tanto ao nível de *hardware* como de *software*.

É importante haver boas ferramentas de trabalho, como tal, a proposta apresentada pretende ser uma mais-valia na criação de bibliotecas de gestos.

Nesta dissertação tentou-se criar uma nova ferramenta para o programador, consistindo no:

- Auxílio na criação de novos gestos.
- Método de reconhecimento para os gestos criados.

Como “intermediário” entre a criação e deteção de gestos foram utilizadas equações. A utilização das equações pretende definir o gesto de forma contínua e simplificar a quantidade de informação a guardar. Este tipo de dados apresentou alguma dificuldade na preparação da informação disponibilizada pelo dispositivo Kinect. Foi necessário desenvolver alguns filtros para se chegar ao resultado pretendido antes de obter as equações.

Foi criado um protótipo para que se pudesse verificar a eficácia. Durante a gravação de alguns gestos notou-se alguma dificuldade na obtenção das equações, muitas dessas dificuldades passaram pela deficiência das coordenadas adquiridas e no caso de alguns gestos mais complexos, pelo defeito já mencionado no capítulo anterior, de utilizar o mesmo conjunto de pontos para a definição das equações $y(x)$ e $z(x)$.

Embora este protótipo não seja uma solução final, mostra uma linha de como pode surgir um produto final para eventuais aplicações comerciais. No entanto, o uso deste protótipo permite já uma boa taxa de reconhecimento.

Com a validação deste método, a nível singular de cada gesto, ficou visível que não existe grande dificuldade em reconhecer um gesto. Este teste pode ser feito em perfeitas condições, tanto a nível de cenário como a nível de processamento por se tratar de um gesto apenas. Já no teste com vários gestos prontos a serem reconhecidos notou-se ligeiramente necessidade de uma maior velocidade de processamento. No entanto este reconhecimento em conjunto apresentou taxas bastante boas.

De salientar ainda que para um melhor reconhecimento as equações devem ser definidas corretamente. Com os ensaios feitos às equações criadas manualmente ficou visível que a eficácia de reconhecimento pode melhorar muito com as equações adequadas.

5.2 Trabalhos Futuros

Como trabalhos futuros, há muitos aspetos que se poderão melhorar. O ponto essencial é resolver o problema de adquirir posições erradas das articulações, esta resolução passa por algum tipo de melhoramento no algoritmo ou então a evolução do *hardware* de forma a garantir mais precisão.

A solução proposta foi pensada de forma genérica e de forma a ser facilmente adaptada a outros dispositivos, seria vantajoso elevar o nível de abstração para uma mais rápida integração em outros sistemas.

No que diz respeito à criação de gestos, foram usadas equações de grau 3 para tentar definir trajetórias complexas, visto ser uma equação com mais zeros (raízes), no entanto durante a criação dos gestos verificou-se que é raro um gesto tomar a trajetória de uma equação de grau 3. Considerando também os testes com as equações criadas manualmente, equações de grau 1 e 2, numa próxima evolução do projeto seria interessante criar uma definição mista, através de equações de grau 2 e retas.

Quanto ao reconhecimento em tempo real, existe sempre alguns melhoramentos a fazer mas o ideal será utilizar processadores de grande frequência, porque o reconhecimento de um grande conjunto de gestos pode ainda não estar preparado para os dias de hoje.

Devido à falta de recursos faltou observar a quantidade de gestos que uma máquina, dos dias de hoje, de elevado processamento conseguiria suportar sem causar atrasos.

Este método poderá ser aplicada/integrada em vários projetos. O mundo dos jogos é sempre de ter em consideração, como por exemplo o Kinect-Sign (Gameiro, Cardoso, & Rybarczyk, 2013) é um jogo aprendizagem de língua gestual portuguesa (LGP). Este jogo é um bom exemplo para a utilização de uma biblioteca de gestos, através da ferramenta apresentada é possível criar os gestos pretendidos para ensinar LGP. Como este, muitos outros projetos podem ser desenvolvidos a partir deste método.

Bibliografia

- Apple. (2013). *Magic Mouse*. Obtido de <http://www.apple.com/magicmouse/>
- Balaji, R., Deepu, V., Madhvanath, S., & Prabhakaran, J. (2006). Handwritten Gesture Recognition for Gesture Keyboard. *Tenth International Workshop on Frontiers in Handwriting Recognition*. La Baule, França.
- Burger, D. (2012). *HomeTechTell Review: Samsung UN55ES8000F 8000 Series LED Smart TV*. Obtido de <http://www.technologytell.com/hometech/85660/hometechtell-review-samsung-un55es8000f-8000-series-led-smart-tv/>
- Chen, W., & Chang, S.-F. (2000). Motion Trajectory Matching of Video Objects. *SPIE/IS&T Storage and Retrieval for Media Databases*. San Jose, CA.
- Ding, Y., Liu, S., Pu, J., & Ni, L. (2013). HUNTS: A Trajectory Recommendation System for Effective and Efficient Hunting of Taxi Passengers. *Mobile Data Management*. Milão, Italia.
- ESPN. (2010). *Donovan gets his game on at UCLA*. Obtido de http://espn.go.com/blog/los-angeles/soccer/post/_id/373/donovan-gets-his-game-on-at-ucla
- Fernandez, D. (2012). *Installing and Using the Kinect Sensor (Beta 2 SDK)*. Obtido de <http://channel9.msdn.com/Series/KinectSDKQuickstarts/Understanding-Kinect-Hardware>
- Flatley, J. (2011). *Visualized: Kinect + night vision = lots and lots and lots of dots*. Obtido de <http://www.engadget.com/2010/11/08/visualized-kinect-night-vision-lots-and-lots-and-lots-of-dots/>
- Gameiro, J., Cardoso, T., & Rybarczyk, Y. (2013). Kinect-Sign, Teaching sign language to "listeners" through a game. *Conference on Electronics, Telecommunications and Computers*. Lisboa, Portugal.
- Hardesty, L. (2012). *MIT news*. Obtido de <http://web.mit.edu/newsoffice/2012/robots-hand-gestures-0314.html>
- Jana, A. (2012). *Kinect for Windows SDK Programming Guide*. Packt Publishing.
- King, A. (2013). *A closer look at why Nintendo is stopping Wii production*. Obtido de <http://wiiudaily.com/2013/10/nintendo-halting-wii-production/>
- Kistler, F. (2011). *FUBI - Full Body Interaction Framework*. Obtido de <http://www.informatik.uni-augsburg.de/en/chairs/hcm/projects/tools/fubi/>
- Kubota, N., & Abe, M. (2006). Human Hand Detection for Gestures Recognition of A Partner Robot. *Automation Congress, 2006. WAC '06. World*. Budapeste, Hungria.
- Lieveld, J. (2013). Obtido de THE FUTURE OF UX/UI: THE NATURAL USER INTERFACE: <http://www.lieveld.com/the-future-of-uxui-the-natural-user-interface/>

- Microsoft. (2013). *Develop with Kinect for Windows*. Obtido de <http://www.microsoft.com/en-us/kinectforwindows/develop/>
- Microsoft. (2013). *Use the computer without the mouse or keyboard*. Obtido de <http://windows.microsoft.com/en-us/windows-vista/use-the-computer-without-the-mouse-or-keyboard-alternative-input-devices>
- Navarro, G. (2004). Pattern Matching. *Journal of Applied Statistics*.
- Noessel, C. (2013). *What Sci-Fi Tells Interaction Designers About Gestural Interfaces*. Obtido de <http://uxdesign.smashingmagazine.com/2013/03/01/sci-fi-interaction-designers-gestural-interfaces/>
- Nomm, S., Leibak, A., Vain, J., & Miyawaki, F. (2010). Application of Volume Bounding Box Decomposition for Surgeon's Hand Gestures Recognition. *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. Glendale, EUA.
- OpenNI. (2013). *OpenNI*. Obtido de <http://www.openni.org/organization/>
- Pu, Q., Gupta, S., Gollakota, S., & Patel, S. (2013). Obtido de Whole-Home Gesture Recognition Using Wireless: <http://wisee.cs.washington.edu/>
- Pu, Q., Gupta, S., Gollakota, S., & Patel, S. (2013). Whole-Home Gesture Recognition Using Wireless Signals. *The 19th Annual International Conference on Mobile Computing and Networking*. Miami, Florida.
- Raiten, S. (2011). *Kinect – Getting Started – Become The Incredible Hulk*. Obtido de <http://www.codeproject.com/Articles/213034/Kinect-Getting-Started-Become-The-Incredible-Hulk>
- Samsung. (2013). *Smart TV*. Obtido de <http://www.samsung.com/us/2013-smart-tv/>
- Samsung. (2013). *Tips & Tricks: Galaxy S 4*. Obtido de <http://www.samsung.com/us/article/tips--tricks-galaxy-s-4>
- Sanders, K. (2013). *Wi-Fi Network WiSee Gesture Control Commanded*. Obtido de <http://www.thevoltreport.com/wi-fi-network-wisee-gesture-control-commanded/>
- Schlomer, T., Benjamin, P., Henze, N., & Susanne, B. (2008). Gesture Recognition with a Wii Controller. *TEI 2008 - Proceedings of the 2nd International Conference on Tangible and Embedded Interaction*. Bonn, Germany.
- Sedgewick, R. (2007). *Department of Computer Science*. Obtido de <http://www.cs.princeton.edu/~rs/AlgsDS07/21PatternMatching.pdf>
- Song, Y., Demirdjian, D., & Davis, R. (2012). Continuous Body and Hand Gesture Recognition for Natural Human-Computer Interaction. *ACM Transactions on Interactive Intelligent Systems*.
- Sony. (2013). *This Is How I Move*. Obtido de <http://us.playstation.com/ps3/playstation-move/>

- Techcyn. (2010). *KINECT™ for XBOX 360®*. Obtido de <http://www.techcyn.com/gadget.php?id=g1&issue=1>
- Thalmic Labs. (2013). *Myo - Gesture control armband*. Obtido de <https://www.thalmic.com/en/myo/>
- Thornton, C. (2013). *Myo Armbands from Thalmic Labs*. Obtido de <http://gadgetgrave.com/myo-armbands-thalmic-labs/>
- Wagner, J., Lingenfelser, F., Baur, T., Damian, L., Kistler, F., & André, E. (2013). The Social Signal Interpretation (SSI) Framework. *ACM International Conference on Multimedia*. Barcelona, Espanha.
- Wixon, D., & Wigdor, D. (2011). *Brave NUI World*. Elsevier Inc.
- Zhou, M., Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., & Chen, X. (2013). Sign Language Recognition and Translation with Kinect. *AFGR 2013 : 10 th IEEE International Conference on Automatic Face and Gesture Recognition*. Shanghai, China.